



УНИВЕРЗИТЕТ У НОВОМ САДУ  
ФАКУЛТЕТ ТЕХНИЧКИХ НАУКА У  
НОВОМ САДУ



мр Небојша Николић

# РАЗВОЈ МЕТОДА ДИЈАГНОСТИКЕ УСИСНОГ СИСТЕМА МОТОРА СА УНУТРАШЊИМ САГОРЕВАЊЕМ

ДОКТОРСКА ДИСЕРТАЦИЈА

Нови Сад, 2015



УНИВЕРЗИТЕТ У НОВОМ САДУ • ФАКУЛТЕТ ТЕХНИЧКИХ НАУКА  
21000 НОВИ САД, Трг Доситеја Обрадовића 6

## КЉУЧНА ДОКУМЕНТАЦИЈСКА ИНФОРМАЦИЈА

Редни број, <b>РБР:</b>	
Идентификациони број, <b>ИБР:</b>	
Тип документације, <b>ТД:</b>	Монографска публикација
Тип записа, <b>ТЗ:</b>	Текстуални штампани материјал
Врста рада, <b>ВР:</b>	Докторска дисертација
Аутор, <b>АУ:</b>	мр Небојша Николић
Ментор, <b>МН:</b>	Проф. др Иван Клинар
Наслов рада, <b>НР:</b>	Развој метода дијагностике усисног система мотора са унутрашњим сагоревањем
Језик публикације, <b>ЈП:</b>	Српски
Језик извода, <b>ЈИ:</b>	Српски / Енглески
Земља публикавања, <b>ЗП:</b>	Република Србија
Уже географско подручје, <b>УГП:</b>	АП Војводина
Година, <b>ГО:</b>	2015
Издавач, <b>ИЗ:</b>	Ауторски репринт
Место и адреса, <b>МА:</b>	21000 Нови Сад, Трг Доситеја Обрадовића 6
Физички опис рада, <b>ФО:</b> (поглавља/страна/ цитата/табела/слика/графика/прилога)	10/244/122/36/152/0/5
Научна област, <b>НО:</b>	Машинско инжењерство
Научна дисциплина, <b>НД:</b>	Мотори са унутрашњим сагоревањем
Предметна одредница/Кључне речи, <b>ПО:</b>	Мотор СУС, усисни систем, детектовање и дијагностиковање неисправности
<b>УДК</b>	
Чува се, <b>ЧУ:</b>	Библиотека Факултета техничких наука
Важна напомена, <b>ВН:</b>	
Извод, <b>ИЗ:</b>	У раду је развијен један математички модел за симулирање понашања неких важних радних параметара мотора СУС, када у његовом усисном систему постоје неисправности типа: „непредвиђени улаз ваздуха у усисни колектор“, „погрешно читавање сензора масеног протока ваздуха“, „погрешно читавање сензора притиска у усисном колектору“, „погрешно читавање сензора температуре у усисном колектору“ и „умањен EGR проток“. На основу резултата овог модела предложен је нови дијагностички концепт, у оквиру којег је развијен један модел за препознавање поменутих неисправности. Предложени концепт је проверен на реалним подацима, прикупљеним испитивањем једног стварног мотора у лабораторијским условима, при чему су добијени задовољавајући резултати.
Датум прихватања теме, <b>ДП:</b>	24.04.2008.
Датум одбране, <b>ДО:</b>	
Чланови комисије, <b>КО:</b>	Председник: Проф. др Стеван Станковски
	Члан: Проф. др Снежана Петковић
	Члан: Ван. проф. др Лазар Савин
	Члан: Доц. др Драган Ружић
	Члан: Доц. др Јован Дорић
	Члан, ментор: Проф. др Иван Клинар
	Потпис ментора



## KEY WORDS DOCUMENTATION

Accession number, <b>ANO</b> :	
Identification number, <b>INO</b> :	
Document type, <b>DT</b> :	Monographic publication
Type of record, <b>TR</b> :	Textual printed document
Contents code, <b>CC</b> :	PhD Thesis
Author, <b>AU</b> :	Nebojša Nikolić
Mentor, <b>MN</b> :	Professor Ivan Klinar, PhD
Title, <b>TI</b> :	Development of an IC Engine Intake Air Path Fault Diagnosis Method
Language of text, <b>LT</b> :	Serbian
Language of abstract, <b>LA</b> :	Serbian / English
Country of publication, <b>CP</b> :	Republic of Serbia
Locality of publication, <b>LP</b> :	AP of Vojvodina
Publication year, <b>PY</b> :	2015
Publisher, <b>PB</b> :	Author's reprint
Publication place, <b>PP</b> :	21000 Novi Sad, Trg Dositeja Obradovića 6,
Physical description, <b>PD</b> : (chapters/pages/ref./tables/pictures/graphs/appendixes)	10/244/122/36/152/0/5
Scientific field, <b>SF</b> :	Mechanical Engineering
Scientific discipline, <b>SD</b> :	Internal combustion engines
Subject/Key words, <b>S/KW</b> :	IC Engine, intake air path, fault detection and diagnosis
<b>UC</b>	
Holding data, <b>HD</b> :	Library of the Faculty of Technical Sciences, Novi Sad
Note, <b>N</b> :	
Abstract, <b>AB</b> :	<p>A mathematical model capable of simulating some important IC engine operating parameters behavior when a fault in its intake air path exists. The faults considered are of the following types: „air leakage in the intake path“, „faulty mass air flow sensor“, „faulty manifold absolute pressure sensor“, „faulty intake air temperature sensor“ and „clogged EGR pipe“. Relying on the data obtained by the fault simulator, a novel diagnosis concept is proposed. A model for fault detection and diagnosis was developed in the scope of the concept. The proposed concept was tested on the real data collected from an automobile IC engine in the laboratory conditions and satisfying results were obtained.</p>
Accepted by the Scientific Board on, <b>ASB</b> :	April 24 <sup>th</sup> , 2008
Defended on, <b>DE</b> :	
Defended Board, <b>DB</b> :	
President:	Professor Stevan Stankovski, PhD
Member:	Professor Snežana Petković, PhD
Member:	Associate professor Lazar Savin, PhD
Member:	Assistant professor Dragan Ružić, PhD
Member:	Assistant professor Jovan Dorić, PhD
Member, Mentor:	Professor Ivan Klinar, PhD
	Mentor's sign

# SADRŽAJ

<b>1</b>	<b>UVOD .....</b>	<b>1</b>
1.1	Predmet istraživanja.....	1
1.2	Cilj istraživanja i hipoteza.....	3
1.3	Struktura disertacije.....	4
1.4	Značaj i doprinos disertacije.....	5
<b>2</b>	<b>PREGLED STANJA U OBLASTI DIJAGNOSTIKE NEISPRAVNOSTI U USISNOM SISTEMU MOTORA.....</b>	<b>8</b>
2.1	Terminologija.....	8
2.2	Uopšte o detektovanju i dijagnostikovanju neispravnosti kod automatski upravljanih sistema .....	9
2.3	Pristupi u detektovanju i dijagnostikovanju neispravnosti .....	11
2.3.1	Tradicionalni (nemodelski) pristup.....	12
2.3.2	Modelski pristup.....	13
2.4	Pregled relevantnih radova u oblasti modelskih metoda za dijagnostikovanje neispravnosti u usisnom sistemu motora.....	17
2.4.1	Radovi koji ilustruju razvoj i primenu analitičkih modela.....	17
2.4.2	Radovi koji ilustruju razvoj i primenu modela zasnovanih na neuronskim mrežama .....	19
2.4.3	Ostali radovi od značaja za predmetno istraživanje.....	21
<b>3</b>	<b>PRIMENA VEŠTAČKIH NEURONSKIH MREŽA U PREPOZNAVANJU UZORAKA.....</b>	<b>24</b>
3.1	Uvodna razmatranja.....	24
3.2	Matematički model neurona.....	26
3.3	Struktura neuronske mreže bez povratnih sprega .....	29
3.4	Proces obuke neuronske mreže .....	31
3.4.1	BP algoritam – teorijske osnove.....	33
3.4.2	Obuka neuronske mreže korišćenjem BP algoritma .....	36
3.5	Postupak razvoja neuronske mreže za prepoznavanje uzoraka .....	38
<b>4</b>	<b>OSREDNJNI MODEL MOTORA .....</b>	<b>43</b>
4.1	Upravljački orijentisani modeli motora .....	43
4.2	Osrednjeni modeli motora.....	45
4.3	Hendriksov osrednjeni model motora .....	47
4.3.1	Podsistem proticanja goriva kroz usisni sistem .....	49
4.3.2	Podsistem dinamike strujanja kroz usisni kolektor .....	51
4.3.3	Podsistem dinamike kolenastog vratila .....	56
<b>5</b>	<b>SIMULATOR NEISPRAVNOSTI U USISNOM SISTEMU .....</b>	<b>60</b>
5.1	Uvod .....	60
5.2	Struktura simulatora neispravnosti.....	61
5.3	Modelovanje simulatora neispravnosti.....	62
5.3.1	Modelovanje snage potrebne za savladavanje otpora na zamajcu .....	63

5.3.2	Modelovanje uticaja nezaptivenosti usisnog sistema.....	65
5.3.2.1	Uticaj na temperaturu u usisnom kolektoru.....	66
5.3.2.2	Uticaj na pritisak u usisnom kolektoru.....	69
5.3.3	Modelovanje pogrešnog očitavanja senzora pritiska u usisnom kolektoru.....	69
5.3.3.1	Motori bez masenog protokomera vazduha.....	69
5.3.3.2	Motori sa masenim protokomerom vazduha.....	70
5.3.4	Modelovanje pogrešnog očitavanja senzora temperature u usisnom kolektoru.....	70
5.3.5	Modelovanje pogrešnog očitavanja senzora masenog protoka vazduha.....	71
5.3.6	Modelovanje umanjenog EGR protoka.....	71
5.3.7	Modelovanje doziranja goriva u slučaju neispravnosti u usisnom sistemu.....	72
5.3.7.1	Motor sa masenim protokomerom, otvorena sprega sa $\lambda$ -sondom.....	73
5.3.7.2	Motor bez protokomera, otvorena sprega sa $\lambda$ -sondom.....	74
5.3.7.3	Rad motora u zatvorenoj sprezi sa $\lambda$ -sondom.....	76
5.4	Primena simulatora neispravnosti.....	79
5.4.1	Ulazni podaci modela <i>NSimulator</i> .....	79
5.4.2	Struktura matrice izlaznih podataka simulatora neispravnosti.....	83
5.4.3	Početni uslovi za simulaciju i veličina matrice izlaznih podataka.....	85
5.4.4	Algoritam programa <i>NSimulatorMenadzer</i> .....	89
5.5	Rezultati primene simulatora neispravnosti.....	94
5.5.1	Motor sa masenim protokomerom vazduha.....	96
5.5.2	Motor bez protokomera vazduha.....	108
5.5.3	Zaključni komentar rezultata simulacija neispravnosti.....	120
<b>6</b>	<b>DIJAGNOSTIČKI MODEL BAZIRAN NA REZULTATIMA SIMULACIJA.....</b>	<b>121</b>
6.1	Uvodne napomene.....	121
6.2	Struktura i koncept funkcionisanja modela.....	121
6.3	Razvoj neuronskih mreža za dijagnostiku neispravnosti u usisnom sistemu.....	124
6.3.1	Definisanje ulaznih i izlaznih veličina.....	124
6.3.2	Prikupljanje i priprema podataka za obuku.....	127
6.3.3	Strukturisanje neuronske mreže.....	133
6.3.4	Obuka neuronske mreže.....	134
6.3.5	Testiranje neuronske mreže.....	135
6.3.6	Računarski programi za praktičnu realizaciju dijagnostičkog modela.....	141
6.3.7	Rezultati testiranja dijagnostičkog modela i diskusija.....	145
6.3.8	Zaključni komentar rezultata testiranja.....	165
<b>7</b>	<b>DIJAGNOSTIČKI MODEL ZASNOVAN NA LABORATORIJSKI DOBIJENIM PODACIMA.....</b>	<b>167</b>
7.1	Uvodne napomene.....	167
7.2	Opis eksperimentalnog postrojenja u celini.....	168
7.3	Opis pojedinih komponenti eksperimentalnog postrojenja.....	171
7.3.1	Motor.....	171
7.3.2	Dinamometar SCHENCK W230.....	173

7.3.3	Ostala merno-akviziciona oprema .....	175
7.4	Realizacija eksperimentalnog dela istraživanja.....	179
7.4.1	Opis izazivanja neispravnosti u usisnom sistemu.....	179
7.4.2	Određivanje baznih radnih tačaka motora .....	182
7.4.3	Postupak snimanja podataka za jedno stanje usisnog sistema.....	184
7.5	Razvoj laboratorijskog dijagnostičkog modela usisnog sistema .....	185
7.5.1	Definisanje ulaznih i izlaznih veličina .....	185
7.5.2	Prikupljanje i priprema podataka za obuku .....	186
7.5.3	Strukturisanje i obuka neuronske mreže .....	192
7.5.4	Testiranje neuronske mreže.....	192
7.5.5	Rezultati testiranja neuronske mreže i diskusija .....	194
7.5.6	Zaključni komentar eksperimentalnih rezultata .....	201
<b>8</b>	<b>ZAKLJUČAK I SMERNICE DALJEG ISTRAŽIVANJA .....</b>	<b>203</b>
<b>9</b>	<b>LITERATURA .....</b>	<b>206</b>
<b>10</b>	<b>PRILOZI .....</b>	<b>213</b>
<b>P1</b>	<b>Najvažniji moduli Simulink modela NSimulator .....</b>	<b>214</b>
<b>P2</b>	<b>Listing izvornog koda programa NSimulatorMenadzer i pratećih programa.....</b>	<b>220</b>
P2.1	NSimulatorMenadzer – izvorni kod.....	220
P2.2	generisiPulsacije – izvorni kod.....	230
P2.3	generisiPulsacijeProc – izvorni kod .....	230
P2.4	iscrtajDijagrame – izvorni kod .....	230
P2.5	oznaciStanja – izvorni kod.....	233
<b>P3</b>	<b>Izvorni kodovi funkcija napisanih u okviru razvoja simulacionog dijagnostičkog modela.....</b>	<b>235</b>
P3.1	MatricaZeljenihIzlaza – izvorni kod .....	235
P3.2	generisiStrukturuZaObuku – izvorni kod.....	235
P3.3	kreirajNetStrukturu – izvorni kod.....	236
P3.4	generisiTestStrukturu – izvorni kod .....	237
P3.5	testirajNetStrukturu – izvorni kod.....	237
P3.6	preradiRedukujNNizlaz – izvorni kod.....	239
<b>P4</b>	<b>Izvorni kodovi funkcija napisanih u svrhu vizuelnog prikaza rezultata ..</b>	<b>240</b>
P4.1	iscrtajNNizlaz – izvorni kod .....	240
P4.2	oznaciStanjaBoje – izvorni kod .....	241
P4.3	stanjaNN – izvorni kod .....	241
<b>P5</b>	<b>Izvorni kodovi programa napisanih u okviru razvoja laboratorijskog dijagnostičkog modela.....</b>	<b>243</b>
P5.1	radneTackeMotora – izvorni kod.....	243
P5.2	grupisaneNeispravnosti – izvorni kod .....	244

## LISTA SLIKA

Sl. 2.1 Razvoj otkaza i zastoja u radu iz neispravnosti koja uzrokuje postepenu ili iznenadnu promenu nekog karakterističnog svojstva sistema [2].....	9
Sl. 2.2 Oblasti funkcionisanja postrojenja sa upravljanjem tolerantnim na neispravnosti [3] .....	10
Sl. 2.3 Šema automatski upravljano (nadgledano) sistema .....	11
Sl. 2.4 Opšta šema modelskog dijagnostičkog sistema [17].....	15
Sl. 3.1 Ilustracija najčešćih zadataka neuronskih mreža: (a) Aproksimacija funkcija; (b) Prepoznavanje uzoraka; (c) Prognoziranje trenda; (d) Kategorizacija uzoraka; (e) Analiza signala u vremenu. [53] ...	25
Sl. 3.2 Šematski prikaz modela neurona .....	26
Sl. 3.3 Šematski prikaz topologije jedne potpuno povezane višeslojne neuronske mreže bez povratnih sprega.....	29
Sl. 3.4 Šematski prikaz nadgledanog učenja neuronske mreže.....	32
Sl. 3.5 Šema BP neuronske mreže sa jednim skrivenim slojem .....	33
Sl. 3.6 Uticaj promene težinskog koeficijenta na funkciju greške.....	35
Sl. 3.7 Faze razvoja neuronske mreže.....	39
Sl. 4.1 Testiranje ECU motora pomoću Hardware-in-the-Loop simulacije [74].....	44
Sl. 4.2 Maseni protok vazduha pored leptira po osrednjenom i ciklusnom modelu motora [30].....	46
Sl. 4.3 Ulazne i izlazne veličine Hendriksovog modela.....	48
Sl. 4.4 Uprošćeni prikaz jednog oto motora sa istaknutim podsistemima Hendriksovog osrednjenog modela .....	48
Sl. 4.5 Šematski prikaz dinamike strujanja goriva kroz usisni sistem .....	49
Sl. 4.6 Promene protoka vazduha oko leptira i pritiska u usisnom kolektoru sa povećanjem otvora leptira za jedan V8 motor, radne zapremine 5 dm <sup>3</sup> [95].....	52
Sl. 4.7 Model strujanja oko leptira pomoću dva odvojena toka vazduha [94].....	53
Sl. 4.8 Primeri zavisnosti veličine ( $\eta v \cdot pi$ ) od $pi$ za različite motore, pri različitim brojevima obrtaja [92].....	55
Sl. 4.9 Izmerene vrednosti i vrednosti dobijene pomoću Hendriksovog osrednjenog modela, za neke bitnije veličine u usisnom sistemu, pri ustaljenom i neustaljenom radu motora [80].....	58
Sl. 5.1 Struktura simulatora neispravnosti, sa ulaznim i izlaznim veličinama.....	62
Sl. 5.2 Sile koje deluju na vozilo u pravcu kretanja [105].....	63
Sl. 5.3 Maseni protoci kroz usisni sistem pri postojanju neželjenog otvora iza leptira .....	65
Sl. 5.4 Algoritam određivanja konačne korigovane vrednosti koeficijenta viška vazduha, $\lambda_{kon}$ .....	78
Sl. 5.5 Generisanje izlazne matrice simulatora neispravnosti.....	84
Sl. 5.6 Ilustracija dostizanja ustaljenog broja obrtaja pri simulacijama tri različite neispravnosti: (a) kad simulacije traju po 8 sekundi; (b) kad simulacije traju po 25 sekundi.....	86
Sl. 5.7 Ilustracija promene karakterističnih veličina tokom ustaljenog rada motora: (a) broj obrtaja motora; (b) pritisak u usisnom kolektoru .....	88
Sl. 5.8 Primer izlaznog signala masenog protokomera vazduha na ustaljenom režimu rada motora [115] .....	88
Sl. 5.9 Logika izvršavanja programa <b>NSimulatorMenadzer</b> .....	89
Sl. 5.10 Koraci zvanične simulacije .....	90
Sl. 5.11 Promena broja obrtaja motora pri simulacijama neispravnosti kod motora sa masenim protokomerom vazduha – Slučaj 1.1 .....	96
Sl. 5.12 Promena pritiska u usisnom kolektoru pri simulacijama neispravnosti kod motora sa masenim protokomerom vazduha – Slučaj 1.1.....	96
Sl. 5.13 Promena temperature u usisnom kolektoru pri simulacijama neispravnosti kod motora sa masenim protokomerom vazduha – Slučaj 1.1.....	97
Sl. 5.14 Promena masenog protoka pored leptira pri simulacijama neispravnosti kod motora sa masenim protokomerom vazduha – Slučaj 1.1.....	97
Sl. 5.15 Karakteristične vrednosti koeficijenta viška vazduha pri simuliranju neispravnosti kod motora sa masenim protokomerom vazduha – Slučaj 1.1.....	99
Sl. 5.16 Relevantni maseni protoci pri simuliranju neispravnosti kod motora sa masenim protokomerom vazduha – Slučaj 1.1 .....	99
Sl. 5.17 Promena broja obrtaja motora pri simulacijama neispravnosti kod motora sa masenim protokomerom vazduha – Slučaj 1.2.....	101
Sl. 5.18 Promena pritiska u usisnom kolektoru pri simulacijama neispravnosti kod motora sa masenim protokomerom vazduha – Slučaj 1.2.....	101
Sl. 5.19 Promena temperature u usisnom kolektoru pri simulacijama neispravnosti kod motora sa masenim protokomerom vazduha – Slučaj 1.2.....	102
Sl. 5.20 Promena masenog protoka pored leptira pri simulacijama neispravnosti kod motora sa masenim protokomerom vazduha – Slučaj 1.2.....	102

Sl. 5.21 Karakteristične vrednosti koeficijenta viška vazduha pri simuliranju neispravnosti kod motora sa masenim protokomerom vazduha– Slučaj 1.2.....	103
Sl. 5.22 Relevantni maseni protoci pri simuliranju neispravnosti kod motora sa masenim protokomerom vazduha– Slučaj 1.2 .....	103
Sl. 5.23 Karakteristične vrednosti koeficijenta viška vazduha pri simuliranju neispravnosti kod motora sa masenim protokomerom vazduha– Slučaj 1.3.....	104
Sl. 5.24 Promena broja obrtaja motora pri simulacijama neispravnosti kod motora sa masenim protokomerom vazduha – Slučaj 1.3 .....	104
Sl. 5.25 Promena pritiska u usisnom kolektoru pri simulacijama neispravnosti kod motora sa masenim protokomerom vazduha – Slučaj 1.3 .....	105
Sl. 5.26 Promena temperature u usisnom kolektoru pri simulacijama neispravnosti kod motora sa masenim protokomerom vazduha – Slučaj 1.3.....	105
Sl. 5.27 Promena masenog protoka pored leptira pri simulacijama neispravnosti kod motora sa masenim protokomerom vazduha – Slučaj 1.3.....	105
Sl. 5.28 Relevantni maseni protoci pri simuliranju neispravnosti kod motora sa masenim protokomerom vazduha– Slučaj 1.3 .....	107
Sl. 5.29 Promena broja obrtaja motora pri simulacijama neispravnosti kod motora bez protokomera vazduha – Slučaj 2.1 .....	108
Sl. 5.30 Promena pritiska u usisnom kolektoru pri simulacijama neispravnosti kod motora bez protokomera vazduha – Slučaj 2.1.....	108
Sl. 5.31 Promena temperature u usisnom kolektoru pri simulacijama neispravnosti kod motora bez protokomera vazduha – Slučaj 2.1 .....	109
Sl. 5.32 Karakteristične vrednosti koeficijenta viška vazduha pri simuliranju neispravnosti kod motora bez protokomera vazduha – Slučaj 2.1 .....	110
Sl. 5.33 Relevantni maseni protoci pri simuliranju neispravnosti kod motora bez protokomera vazduha – Slučaj 2.1 .....	110
Sl. 5.34 Promena broja obrtaja motora pri simulacijama neispravnosti kod motora bez protokomera vazduha – Slučaj 2.2 .....	113
Sl. 5.35 Promena pritiska u usisnom kolektoru pri simulacijama neispravnosti kod motora bez protokomera vazduha – Slučaj 2.2.....	113
Sl. 5.36 Promena temperature u usisnom kolektoru pri simulacijama neispravnosti kod motora bez protokomera vazduha – Slučaj 2.2 .....	114
Sl. 5.37 Karakteristične vrednosti koeficijenta viška vazduha pri simuliranju neispravnosti kod motora bez protokomera vazduha – Slučaj 2.2 .....	115
Sl. 5.38 Relevantni maseni protoci pri simuliranju neispravnosti kod motora bez protokomera vazduha – Slučaj 2.2 .....	115
Sl. 5.39 Karakteristične vrednosti koeficijenta viška vazduha pri simuliranju neispravnosti kod motora bez protokomera vazduha – Slučaj 2.3 .....	116
Sl. 5.40 Promena broja obrtaja motora pri simulacijama neispravnosti kod motora bez protokomera vazduha – Slučaj 2.3 .....	117
Sl. 5.41 Promena pritiska u usisnom kolektoru pri simulacijama neispravnosti kod motora bez protokomera vazduha – Slučaj 2.3.....	117
Sl. 5.42 Promena temperature u usisnom kolektoru pri simulacijama neispravnosti kod motora bez protokomera vazduha – Slučaj 2.3 .....	118
Sl. 5.43 Relevantni maseni protoci pri simuliranju neispravnosti kod motora bez protokomera vazduha – Slučaj 1.3 .....	119
Sl. 6.1 Blok-šema principa rada dijagnostičkog modela.....	122
Sl. 6.2 Ilustracija matrice željenih izlaza neuronske mreže.....	130
Sl. 6.3 Prozor za izbor Excel-fajlova pri generisanju skupa za obuku neuronske mreže.....	132
Sl. 6.4 Primer automatski generisanog naziva fajla za obuku neuronske mreže .....	133
Sl. 6.5 Primer vizuelnog prikaza rezultata testiranja neuronske mreže.....	142
Sl. 6.6 Primer vizuelnog prikaza prerađenih izlaza neuronske mreže .....	144
Sl. 6.7 Rezultati testiranja neuronske mreže – motor sa masenim protokomerom vazduha, zatvorena sprega sa $\lambda$ -sondom, V stepen prenosa, otvor leptira 17%, temperatura okoline 30°C, greška $1.7783 \cdot 10^{-7}$ .....	146
Sl. 6.8 Rezultati testiranja neuronske mreže – motor sa masenim protokomerom vazduha, otvorena sprega sa $\lambda$ -sondom, III stepen prenosa, otvor leptira 7%, temperatura okoline 20°C, greška $2.0731 \cdot 10^{-7}$ .....	146
Sl. 6.9 Rezultati testiranja neuronske mreže – motor bez protokomera vazduha, zatvorena sprega sa $\lambda$ -sondom, IV stepen prenosa, otvor leptira 11%, temperatura okoline 0°C, greška $3.4562 \cdot 10^{-7}$ .....	147



Sl. 6.10	Rezultati testiranja neuronske mreže – motor bez protokomera vazduha, otvorena sprega sa $\lambda$ -sondom, II stepen prenosa, otvor leptira 6%, temperatura okoline 10°C, greška $1.866 \cdot 10 - 7$ .....	147
Sl. 6.11	Rezultati testiranja neuronske mreže – motor sa masenim protokomerom vazduha, zatvorena sprega sa $\lambda$ -sondom, IV stepen prenosa, otvor leptira 8.8%, temperatura okoline 0°C, greška $1.8881 \cdot 10 - 7$ .....	148
Sl. 6.12	Rezultati testiranja neuronske mreže – motor sa masenim protokomerom vazduha, otvorena sprega sa $\lambda$ -sondom, II stepen prenosa, otvor leptira 7.2%, temperatura okoline 28°C, greška 0.0018526 (a) na bazi trenutnih vrednosti; (b) na bazi uprosečenih vrednosti.....	149
Sl. 6.13	Rezultati testiranja neuronske mreže – motor bez protokomera vazduha, zatvorena sprega sa $\lambda$ -sondom, V stepen prenosa, otvor leptira 12%, temperatura okoline –5°C, greška $6.8749 \cdot 10 - 6$ ....	150
Sl. 6.14	Rezultati testiranja neuronske mreže – motor bez protokomera vazduha, otvorena sprega sa $\lambda$ -sondom, III stepen prenosa, otvor leptira 7.3%, temperatura okoline –5°C, greška $3.1834 \cdot 10 - 7$ ..	150
Sl. 6.15	Rezultati testiranja neuronske mreže – motor sa masenim protokomerom vazduha, zatvorena sprega sa $\lambda$ -sondom, V stepen prenosa, otvor leptira 17%, temperatura okoline 30°C, greška $1.7783 \cdot 10 - 7$ , nepoznata stanja grupe 1; (a) na bazi trenutnih vrednosti, (b) na bazi uprosečenih vrednosti.....	152
Sl. 6.16	Rezultati testiranja neuronske mreže – motor sa masenim protokomerom vazduha, zatvorena sprega sa $\lambda$ -sondom, IV stepen prenosa, otvor leptira 13%, temperatura okoline 10°C, greška $1.8881 \cdot 10 - 7$ , nepoznata stanja grupe 2.....	153
Sl. 6.17	Rezultati testiranja neuronske mreže – motor sa masenim protokomerom vazduha, otvorena sprega sa $\lambda$ -sondom, III stepen prenosa, otvor leptira 7%, temperatura okoline 20°C, greška $2.0731 \cdot 10 - 7$ , nepoznata stanja grupe 1.....	153
Sl. 6.18	Rezultati testiranja neuronske mreže – motor sa masenim protokomerom vazduha, otvorena sprega sa $\lambda$ -sondom, II stepen prenosa, otvor leptira 6%, temperatura okoline 10°C, greška 0.0018526 nepoznata stanja grupe 2.....	154
Sl. 6.19	Rezultati testiranja neuronske mreže – motor bez protokomera vazduha, zatvorena sprega sa $\lambda$ -sondom, V stepen prenosa, otvor leptira 13%, temperatura okoline 20°C, greška $6.8749 \cdot 10 - 7$ , nepoznata stanja grupe 1 .....	154
Sl. 6.20	Rezultati testiranja neuronske mreže – motor bez protokomera vazduha, zatvorena sprega sa $\lambda$ -sondom, II stepen prenosa, otvor leptira 5%, temperatura okoline 0°C, greška $2.4946 \cdot 10 - 7$ , nepoznata stanja grupe 2 .....	155
Sl. 6.21	Rezultati testiranja neuronske mreže – motor bez protokomera vazduha, otvorena sprega sa $\lambda$ -sondom, III stepen prenosa, otvor leptira 8%, temperatura okoline 30°C, greška $3.1834 \cdot 10 - 7$ , nepoznata stanja grupe 1 .....	155
Sl. 6.22	Rezultati testiranja neuronske mreže – motor bez protokomera vazduha, otvorena sprega sa $\lambda$ -sondom, IV stepen prenosa, otvor leptira 11%, temperatura okoline 0°C, greška $3.1709 \cdot 10 - 7$ , nepoznata stanja grupe 2 .....	156
Sl. 6.23	Rezultati testiranja neuronske mreže – motor sa masenim protokomerom vazduha, zatvorena sprega sa $\lambda$ -sondom, V stepen prenosa, otvor leptira 14%, temperatura okoline 12°C, greška $1.7783 \cdot 10 - 7$ , nepoznata stanja grupe 1.....	157
Sl. 6.24	Rezultati testiranja neuronske mreže – motor sa masenim protokomerom vazduha, zatvorena sprega sa $\lambda$ -sondom, IV stepen prenosa, otvor leptira 12.5%, temperatura okoline 28°C, greška $1.8881 \cdot 10 - 7$ , nepoznata stanja grupe 2.....	157
Sl. 6.25	Rezultati testiranja neuronske mreže – motor sa masenim protokomerom vazduha, otvorena sprega sa $\lambda$ -sondom, III stepen prenosa, otvor leptira 7,4%, temperatura okoline 12°C, greška $2.0731 \cdot 10 - 7$ , nepoznata stanja grupe 1.....	158
Sl. 6.26	Rezultati testiranja neuronske mreže – motor sa masenim protokomerom vazduha, otvorena sprega sa $\lambda$ -sondom, II stepen prenosa, otvor leptira 6.7%, temperatura okoline 28°C, greška 0.0018526 nepoznata stanja grupe 2.....	158
Sl. 6.27	Rezultati testiranja neuronske mreže – motor bez protokomera vazduha, zatvorena sprega sa $\lambda$ -sondom, V stepen prenosa, otvor leptira 18%, temperatura okoline 28°C, greška $6.8749 \cdot 10 - 7$ , nepoznata stanja grupe 1 .....	159
Sl. 6.28	Rezultati testiranja neuronske mreže – motor bez protokomera vazduha, zatvorena sprega sa $\lambda$ -sondom, II stepen prenosa, otvor leptira 5.3%, temperatura okoline 12°C, greška $2.4946 \cdot 10 - 7$ , nepoznata stanja grupe 2 .....	159
Sl. 6.29	Rezultati testiranja neuronske mreže – motor bez protokomera vazduha, otvorena sprega sa $\lambda$ -sondom, III stepen prenosa, otvor leptira 8.2%, temperatura okoline 28°C, greška $3.1834 \cdot 10 - 7$ , nepoznata stanja grupe 1 .....	160
Sl. 6.30	Rezultati testiranja neuronske mreže – motor bez protokomera vazduha, otvorena sprega sa $\lambda$ -sondom, IV stepen prenosa, otvor leptira 10.5%, temperatura okoline 12°C, greška $3.1709 \cdot 10 - 7$ , nepoznata stanja grupe 2.....	160

Sl. 6.31 Rezultati testiranja neuronske mreže – zatvorena sprega sa $\lambda$ -sondom, III stepen prenosa, otvor leptira 7.4%, temperatura okoline 12°C, masa vozila 1100 kg, atmosferski pritisak 1 bar, brzina vetra 0 km/h, uzdužni nagib puta 0% .....	161
Sl. 6.32 Rezultati testiranja neuronske mreže – zatvorena sprega sa $\lambda$ -sondom, III stepen prenosa, otvor leptira 7.4%, temperatura okoline 12°C, masa vozila 1000 kg, atmosferski pritisak 1.02 bar, brzina vetra 0 km/h, uzdužni nagib puta 0% .....	162
Sl. 6.33 Rezultati testiranja neuronske mreže – zatvorena sprega sa $\lambda$ -sondom, IV stepen prenosa, otvor leptira 10.5%, temperatura okoline 12°C, masa vozila 1000 kg, atmosferski pritisak 1 bar, brzina vetra +15 km/h, uzdužni nagib puta 0% .....	162
Sl. 6.34 Rezultati testiranja neuronske mreže – zatvorena sprega sa $\lambda$ -sondom, IV stepen prenosa, otvor leptira 10.5%, temperatura okoline 12°C, masa vozila 1000 kg, atmosferski pritisak 1 bar, brzina vetra 0 km/h, uzdužni nagib puta +1%.....	163
Sl. 6.35 Rezultati testiranja neuronske mreže – otvorena sprega sa $\lambda$ -sondom, III stepen prenosa, otvor leptira 7.4%, temperatura okoline 12°C, masa vozila 1000 kg, atmosferski pritisak 0.98 bar, brzina vetra 0 km/h, uzdužni nagib puta 0% .....	163
Sl. 6.36 Rezultati testiranja neuronske mreže – otvorena sprega sa $\lambda$ -sondom, III stepen prenosa, otvor leptira 7.4%, temperatura okoline 12°C, masa vozila 1000 kg, atmosferski pritisak 1 bar, brzina vetra –15 km/h, uzdužni nagib puta 0% .....	164
Sl. 6.37 Rezultati testiranja neuronske mreže – otvorena sprega sa $\lambda$ -sondom, IV stepen prenosa, otvor leptira 7.4%, temperatura okoline 12°C, masa vozila 900 kg, atmosferski pritisak 1 bar, brzina vetra 0 km/h, uzdužni nagib puta 0%.....	164
Sl. 6.38 Rezultati testiranja neuronske mreže – otvorena sprega sa $\lambda$ -sondom, IV stepen prenosa, otvor leptira 7.4%, temperatura okoline 12°C, masa vozila 1000 kg, atmosferski pritisak 1 bar, brzina vetra 0 km/h, uzdužni nagib puta –1%.....	165
Sl. 7.1 Eksperimentalno postrojenje.....	168
Sl. 7.2 Šematski prikaz eksperimentalnog postrojenja: 1 – senzor položaja leptira, 2 – senzor pritiska u usisnom kolektoru, 3 – senzor temperature u usisnom kolektoru, 4 – senzor referentnog položaja bregastog vratila (Hall-ov davač), 5 – enkoder, 6 – senzor temperature rashladne tečnosti, 7 – senzor broja obrtaja, 8 – širokopojasna $\lambda$ -sonda, 9 – senzor sile, 10 – merno pojačalo, 11 – akviziciona kartica, 12 – desktop računar, 13 – VEMS upravljačka jedinica (kontroler), 14 – laptop računar.....	169
Sl. 7.3 Korisnički interfejs LabView aplikacije za praćenje osnovnih parametara režima rada motora.....	170
Sl. 7.4 VEMS upravljačka jedinica (VEMS kontroler).....	172
Sl. 7.5 Korisnički interfejs „VemsTune“-a .....	172
Sl. 7.6 Originalni senzori na motoru EFI 1.1: (a) senzor položaja leptira, (b) senzor broja obrtaja motora, (c) senzor temperature u usisnom kolektoru, (d) senzor temperature rashladne tečnosti.....	173
Sl. 7.7 Komandni orman sa najvažnijim detaljima: 1) točkić za podešavanje intenziteta kočenja, 2) tasteri za izbor režima kočenja, 3) skala broja obrtaja, 4) skala sile kočenja.....	174
Sl. 7.8 Princip određivanja obrtnog momenta motora pomoću dinamometru SCHENCK W230: 1) senzor sile HBM U2, 2) stator dinamomentra 3) poluga statora.....	174
Sl. 7.9 Merno pojačalo KWS /T-5 sa napajanjem EA-PS 2016-100.....	175
Sl. 7.10 Izgled akvizicione kartice NI PCI-6070E.....	176
Sl. 7.11 Izgled montiranog enkodera HEDSS ISC 3806 .....	177
Sl. 7.12 Hall-ov davač Honeywell 1GT101DC: (a) izgled, (b) montiran na motoru.....	177
Sl. 7.13 Senzor apsolutnog pritiska u usisnom kolektoru VEMS MPXH6400.....	178
Sl. 7.14 Širokopojasna $\lambda$ - sonda Bosch LSU 4.2 montirana na motor Yugo EFI 1.1 .....	179
Sl. 7.15 Komponente za izazivanje ulaza „falš“-vazduha: (a) čepovi sa ubušanim otvorima, (b) montirana slavina sa jednim od čepova .....	180
Sl. 7.16 Ilustracija prozora u VemsTune-u, u kojem se može promeniti ofset MAP senzora .....	181
Sl. 7.17 Bazne radne tačke motora Yugo EFI 1.1, za zadate podatke o otporima koje treba da savlada, u različitim stepenima prenosa.....	183
Sl. 7.18 Ilustracija matrice željenih izlaza neuronske mreže .....	188
Sl. 7.19 Rezultati merenja u zimskim mesecima pri približnom otvoru leptira 12%, i momentu kočenja 34 Nm: (a) otvor leptira, (b) broj obrtaja motora, (c) apsolutni pritisak u usisnom kolektoru, (d) temperatura u usisnom kolektoru .....	190
Sl. 7.20 Rezultati merenja u letnjim mesecima pri približnom otvoru leptira 12%, i momentu kočenja 34 Nm: (a) otvor leptira, (b) broj obrtaja motora, (c) apsolutni pritisak u usisnom kolektoru, (d) temperatura u usisnom kolektoru .....	190
Sl. 7.21 Rezultati merenja u zimskim mesecima pri približnom otvoru leptira 20%, i momentu kočenja 45 Nm: (a) otvor leptira, (b) broj obrtaja motora, (c) apsolutni pritisak u usisnom kolektoru, (d) temperatura u usisnom kolektoru .....	191

Sl. 7.22 Rezultati merenja u letnjim mesecima pri približnom otvoru leptira 20%, i momentu kočenja 45 Nm: (a) otvor leptira, (b) broj obrtaja motora, (c) apsolutni pritisak u usisnom kolektoru, (d) temperatura u usisnom kolektoru .....	191
Sl. 7.23 Rezultati testiranja poznatim stanjima u poznatoj baznoj radnoj tački: otvor leptira $\approx 8\%$ , moment kočenja $\approx 26Nm$ , broj obrtaja $\approx 1742 \text{ min} - 1$ .....	195
Sl. 7.24 Rezultati testiranja poznatim stanjima u poznatoj baznoj radnoj tački: otvor leptira $\approx 10\%$ , moment kočenja $\approx 30Nm$ , broj obrtaja $\approx 1948 \text{ min} - 1$ .....	195
Sl. 7.25 Rezultati testiranja poznatim stanjima u poznatoj baznoj radnoj tački: otvor leptira $\approx 12\%$ , moment kočenja $\approx 34Nm$ , broj obrtaja $\approx 2160 \text{ min} - 1$ .....	195
Sl. 7.26 Rezultati testiranja poznatim stanjima u poznatoj baznoj radnoj tački: otvor leptira $\approx 15\%$ , moment kočenja $\approx 37Nm$ , broj obrtaja $\approx 2273 \text{ min} - 1$ .....	196
Sl. 7.27 Rezultati testiranja poznatim stanjima u poznatoj baznoj radnoj tački: otvor leptira $\approx 17\%$ , moment kočenja $\approx 40Nm$ , broj obrtaja $\approx 2392 \text{ min} - 1$ .....	196
Sl. 7.28 Rezultati testiranja poznatim stanjima u poznatoj baznoj radnoj tački: otvor leptira $\approx 20\%$ , moment kočenja $\approx 45Nm$ , broj obrtaja $\approx 2597 \text{ min} - 1$ .....	196
Sl. 7.29 Rezultati testiranja poznatim stanjima u nepoznatoj baznoj radnoj tački: otvor leptira $\approx 18\%$ , moment kočenja $\approx 42Nm$ , broj obrtaja $\approx 2475 \text{ min} - 1$ .....	197
Sl. 7.30 Rezultati testiranja poznatim stanjima u nepoznatoj baznoj radnoj tački: otvor leptira $\approx 19\%$ , moment kočenja $\approx 43Nm$ , broj obrtaja $\approx 2529 \text{ min} - 1$ .....	197
Sl. 7.31 Rezultati testiranja nepoznatim stanjima u poznatoj baznoj radnoj tački: otvor leptira $\approx 8\%$ , moment kočenja $\approx 26Nm$ , broj obrtaja $\approx 1742 \text{ min} - 1$ .....	198
Sl. 7.32 Rezultati testiranja nepoznatim stanjima u poznatoj baznoj radnoj tački: otvor leptira $\approx 10\%$ , moment kočenja $\approx 30Nm$ , broj obrtaja $\approx 1948 \text{ min} - 1$ .....	198
Sl. 7.33 Rezultati testiranja nepoznatim stanjima u poznatoj baznoj radnoj tački: otvor leptira $\approx 12\%$ , moment kočenja $\approx 34Nm$ , broj obrtaja $\approx 2160 \text{ min} - 1$ .....	199
Sl. 7.34 Rezultati testiranja nepoznatim stanjima u poznatoj baznoj radnoj tački: otvor leptira $\approx 15\%$ , moment kočenja $\approx 37Nm$ , broj obrtaja $\approx 2273 \text{ min} - 1$ .....	199
Sl. 7.35 Rezultati testiranja nepoznatim stanjima u poznatoj baznoj radnoj tački: otvor leptira $\approx 17\%$ , moment kočenja $\approx 40Nm$ , broj obrtaja $\approx 2392 \text{ min} - 1$ .....	200
Sl. 7.36 Rezultati testiranja nepoznatim stanjima u poznatoj baznoj radnoj tački: otvor leptira $\approx 20\%$ , moment kočenja $\approx 45Nm$ , broj obrtaja $\approx 2597 \text{ min} - 1$ .....	200
Sl. 7.37 Rezultati testiranja nepoznatim stanjima u nepoznatoj baznoj radnoj tački: otvor leptira $\approx 18\%$ , moment kočenja $\approx 42Nm$ , broj obrtaja $\approx 2475 \text{ min} - 1$ .....	201
Sl. 7.38 Rezultati testiranja nepoznatim stanjima u nepoznatoj baznoj radnoj tački: otvor leptira $\approx 19\%$ , moment kočenja $\approx 43Nm$ , broj obrtaja $\approx 2529 \text{ min} - 1$ .....	201
Sl. 10.1 Šema prvog nivoa kompletnog modela NSimulator .....	214
Sl. 10.2 Izračunavanje pritiska u usisnom kolektoru (modul „d/dt pi“ sa Sl. 10.1).....	214
Sl. 10.3 Određivanje broja obrtaja kolenastog vratila (modul „Crankshaft“ sa Sl. 10.1) .....	215
Sl. 10.4 Izračunavanje temperature u usisnom kolektoru (modul „Ti_calc“ sa Sl. 10.1).....	216
Sl. 10.5 Određivanje masenog protoka „falš“ vazduha (modul „dmleakdt_calc“ sa Sl. 10.2).....	216
Sl. 10.6 Izbor željene vrednosti koeficijenta viška vazduha u zavisnosti od režima rada motora (modul „Izaberi_lambda“ sa Sl. 10.3).....	216
Sl. 10.7 Određivanje potrebne količine goriva pri postojanju neispravnosti u usisnom sistemu (modul „Doziranje_sa_greskom“ sa Sl. 10.3) .....	217
Sl. 10.8 Simuliranje grešaka senzora pritiska i temperature u usisnom kolektoru (modul „PritTempGreska“ sa Sl. 10.7).....	217
Sl. 10.9 Određivanje masenog protoka vazduha kroz usisne kanale, na osnovu (neispravnih ili ispravnih) signala sa senzora (modul „dmacyldt_ECU“ sa Sl. 10.7).....	217
Sl. 10.10 Korigovanje potrebne količine goriva pri radu motora u povratnoj sprezi sa $\lambda$ -sondom (Modul „Korekcija_doze_goriva“ sa Sl. 10.3) .....	218
Sl. 10.11 Korekcija koeficijenta viška vazduha na vrednost koju dozvoljava elektronska upravljačka jedinica (modul „lambda_korekcija“ sa Sl. 10.10).....	218
Sl. 10.12 Struktura modula „smesa_bogatija“ i „smesa_siromasnija“ sa Sl. 10.11.....	218
Sl. 10.13 Određivanje snage potrebne za savladavanje otpora kretanja vozila (modul „Pb“ sa Sl. 10.3) .....	219

## LISTA TABELA

Tab. 1.1 Struktura disertacije .....	4
Tab. 2.1 Izvod iz baze pravila dijagnostičkog sistema predstavljenog u [15].....	14
Tab. 3.1 Najčešći oblici aktivacione funkcije .....	27
Tab. 5.1 Konstantni ulazni podaci korišćeni u virtuelnim eksperimentima.....	79
Tab. 5.2 Parametri za izbor toka simulacije.....	80
Tab. 5.3 Mapa željenih vrednosti koeficijenta viška vazduha.....	81
Tab. 5.4 Podrazumevane vrednosti parametara za izbor neispravnosti .....	81
Tab. 5.5 Vrednosti ključnih parametara pri simulacijama konkretnih neispravnosti.....	82
Tab. 5.6 Neispravnosti u zavisnosti od načina određivanja masenog protoka vazduha .....	83
Tab. 5.7 Ilustracija podataka sadržanih u izlaznoj matrici <i>MatBRT</i> .....	94
Tab. 5.8 Objašnjenje oznaka stanja usisnog sistema.....	95
Tab. 6.1 Izlazne veličine neuronskih mreža – slučaj sa masenim protokomerom vazduha.....	124
Tab. 6.2 Izlazne veličine neuronskih mreža – slučaj bez protokomera vazduha .....	125
Tab. 6.3 Ulazne promenljive neuronskih mreža dijagnostičkog modela .....	127
Tab. 6.4 Plan izvođenja virtuelnih eksperimenata - obuka neuronskih mreža .....	128
Tab. 6.5 Plan izvođenja virtuelnih eksperimenata - testiranje neuronskih mreža (Nivo I) .....	136
Tab. 6.6 Plan izvođenja virtuelnih eksperimenata - testiranje neuronskih mreža (Nivo II).....	136
Tab. 6.7 Pregled simuliranih nepoznatih neispravnosti – grupa 1 .....	137
Tab. 6.8 Pregled simuliranih nepoznatih neispravnosti – grupa 2 .....	138
Tab. 6.9 Plan izvođenja virtuelnih eksperimenata - testiranje neuronskih mreža (Nivo III) .....	138
Tab. 6.10 Plan izvođenja virtuelnih eksperimenata - testiranje neuronskih mreža (Nivo IV).....	139
Tab. 6.11 Plan izvođenja virtuelnih eksperimenata - testiranje neuronskih mreža (Nivo V).....	140
Tab. 7.1 Osnovni tehnički podaci motora YUGO EFI 1.1.....	171
Tab. 7.2 Osnovni tehnički podaci dinamometra SCHENCK W230 .....	175
Tab. 7.3 Tehničke karakteristike kartice NI PCI-6070E .....	176
Tab. 7.4 Osnovne tehničke karakteristike enkodera HEDSS ISC 3806.....	177
Tab. 7.5 Osnovne tehničke karakteristike lambda-sonde Bosch LSU 4.2 .....	178
Tab. 7.6 Oznake i nazivi razmatranih stanja usisnog sistema motora Yugo EFI 1.1 .....	179
Tab. 7.7 Rezultati određivanja parcijalnih brzinskih karakteristika momenta ispitivanog motora.....	182
Tab. 7.8 Bazne radne tačke ispitivanog motora, u kojima će se izazivati neispravnosti .....	184
Tab. 7.9 Ilustracija redukovano skupa podataka snimljenih pri jednom merenju.....	186
Tab. 7.10 Plan izvođenja laboratorijskih eksperimenata – obuka neuronske mreže.....	187
Tab. 7.11 Plan izvođenja eksperimenata – testiranje neuronskih mreža (Nivo II).....	193
Tab. 7.12 Nepoznate neispravnosti izazivane u eksperimentima za testiranje.....	193
Tab. 7.13 Plan izvođenja eksperimenata – testiranje neuronskih mreža (Nivo III).....	193
Tab. 7.14 Plan izvođenja eksperimenata – testiranje neuronskih mreža (Nivo IV).....	194

# 1 UVOD

## 1.1 Predmet istraživanja

U poslednjih nekoliko decenija problem zagađenja čovekove okoline sve više zaokuplja pažnju celokupne svetske javnosti. Osnovni razlog za to je ubrzani tehnološki razvoj koji je praćen intenzivnim povećanjem energetske potrebe čovečanstva. U pokušajima da za svoje potrebe iskoristi što više energije sadržane u prirodnim resursima, čovek sve više negativno utiče na svoju okolinu, pa i pomenuti problem zagađenja postaje sve veći. Značajan udeo u ovom negativnom uticaju na životnu sredinu ima emisija izduvnih gasova motornih vozila. Na prvi pogled, moglo bi se zaključiti da se ovaj udeo smanjuje sa tehnološkim napretkom čovečanstva, imajući u vidu da proizvođači automobila primenjuju sve naprednija tehnološka rešenja, čiji je jedan od osnovnih ciljeva da čovekovu okolinu poštede od štetnog uticaja. Kada se posmatra svako pojedinačno vozilo, ovo jeste tačno, ali treba uzeti u obzir činjenicu da ukupan broj vozila na planeti konstantno raste. U tom slučaju, može se konstatovati da će i briga čovečanstva o zaštiti životne sredine vremenom biti sve veća.

Kada je u pitanju oblast motornih vozila, ispostavilo se da su glavni pokretači tehnološkog razvoja zakonodavci u razvijenim zemljama. Njihovi zahtevi za umanjenoj sadržaja štetnih komponenti izduvne emisije, kao i za umanjenoj prosečne potrošnje goriva na nivou cele države, uticali su na uvođenje elektronskog upravljanja motorima sa unutrašnjim sagorevanjem 70-ih godina prošlog veka. Isti ovi činoci su i glavni razlog za kasnije ustanovljene OBD (On Board Diagnostics) propise koji se odnose na dijagnostičke sposobnosti elektronske upravljačke jedinice motora SUS u motornim vozilima. U svemu ovome prednjačilo se u Sjedinjenim Američkim Državama, tačnije u Kaliforniji, gde su sredinom 90-ih godina uvedeni još stroži OBD II propisi. OBD II propisi morala su da zadovolje sva putnička i laka teretna vozila koja su prodavana u SAD od 1996. godine. Između ostalog, OBD II propisi zahtevaju da se neprekidno nadgleda rad svih komponenti vozila, čiji otkazi mogu dovesti do povećanja emisije izduvnih gasova za preko 50% u odnosu na važeće američke FTP<sup>1</sup> standarde, kao i da na vozilu postoji dijagnostički sistem sa sposobnošću otkrivanja grešaka u radu tih komponenti u realnom vremenu. I u Evropskoj Uniji od 2000. godine svi automobili sa oto motorom, a od 2003. godine i svi automobili sa dizel motorom, moraju da zadovolje EOBD propise (evropska varijanta OBD II propisa).

Da bi neko vozilo zadovoljilo OBD II ili EOBD propise, neophodno je da ono poseduje OBD sistem, koji je u stanju da na vozilu automatski otkriva i identifikuje neispravnosti, obuhvaćene odgovarajućim OBD propisima. OBD sistem nije neki poseban deo na vozilu, nego se u velikoj meri prepliće sa sistemom za upravljanje motorom. Sa jedne strane, može se reći da je OBD sistem integrisan u sistem za upravljanje motorom, a sa druge strane da predstavlja njegovu nadgradnju. Drugim rečima, u sastav OBD sistema ulazi elektronska upravljačka jedinica (ECU) motora i veći broj senzora koji pripadaju sistemu upravljanja motorom, ali OBD sistem može sadržati i neke dodatne senzore, pomoću kojih se mere vrednosti veličina, važnih sa dijagnostičkog aspekta. Osim hardverskog dela OBD sistema, tu je i softverski deo, koji predstavlja nadgradnju upravljačkog softvera, smeštenog u

---

<sup>1</sup> Federal Test Procedure –postupak testiranja izduvne emisije na dinamometarskim valjcima u trajanju od 1874 sekunde, koji simulira urbani vozni ciklus po centru Los Anđelesa

elektronskoj upravljačkoj jedinici motora. Procenjuje se da je čak preko 50% kompletnog softvera u ECU motora posvećeno dijagnostici [1].

Na današnjim automobilima OBD sistem kontinualno nadgleda veliki broj senzora (senzor temperature rashladne tečnosti, senzor temperature vazduha u usisnom sistemu, senzor položaja leptira, senzor apsolutnog pritiska u usisnom kolektoru, senzor masenog protoka vazduha, senzor položaja bregastog vratila, senzor broja obrtaja, senzor brzine vozila itd.), ali i neke aktuatore (EGR ventil, koračni elektromotor u sistemu upravljanja praznim hodom, elektromagnetni ventil u sistemu za upravljanje isparavanjem goriva iz rezervoara) ili druge komponente (sistem paljenja) bitne za održavanje izduvne emisije u dozvoljenim granicama. Na svakoj od pomenutih komponenti može se pojaviti neka neispravnost, ali se isto tako neispravnosti mogu pojaviti i na drugim mestima u njihovim strujnim kolima, kao što su prekidači, releji, provodnici. Zbog toga OBD sistem tokom rada motora vrši testiranje pobrojanih komponenti u električnom smislu, tj. proverava da li negde postoji kratak spoj ili je možda neko strujno kolo u prekidu zbog oštećenja provodnika ili pohabanih kontakata. Ovaj tip neispravnosti se najlakše otkriva, ali postoje i drugačije neispravnosti, koje nije tako lako otkriti.

Može se desiti da u strujnom kolu nekog senzora nema ni kratkih spojeva, ni prekida, ali da ipak postoji neispravnost, kao na primer, kada ECU od senzora prima neprirodne vrednosti signala. Takve vrednosti signala su izvan normalnog opsega vrednosti, u kojem signal ispravnog senzora treba da se kreće. Ove neispravnosti mogu da budu posledica kontaminiranog ili oštećenog senzora ili visoke vrednosti otpora u nekom delu strujnog kola tog senzora. Ovakve neispravnosti je malo teže otkriti, nego one tipa „kratak spoj“ ili „prekid strujnog kola“, ali je to i dalje jednostavan zadatak za današnje OBD sisteme. Naime, vrši se jednostavna provera, da li trenutna vrednost koju daje senzor, pripada predefinisanim intervalima vrednosti, koji je za svaki senzor smešten u memoriji ECU. Međutim, signal nekog senzora može slati elektronskoj upravljačkoj jedinici sasvim prirodnu i normalnu vrednost, a da ona ipak bude pogrešna, u smislu da se malo razlikuje od tačne vrednosti veličine koju taj senzor meri. To može biti posledica kontaminiranog ili pohabanog senzora ili loših električnih kontakata. Veoma je teško otkriti takav tip neispravnosti, kada neki senzor očitava malo višu ili malo nižu vrednost od tačne vrednosti, a još teže je identifikovati koji je senzor u pitanju. Iako su na prvi pogled male, ove neispravnosti mogu da utiču na pogoršanje emisije izduvnih gasova, naročito ako se radi o senzorima koji direktno ili indirektno utiču na obrazovanje smeše. Osim štetnog uticaja na čovekovu okolinu, ovakve neispravnosti mogu da imaju negativne posledice u još nekim, ne manje važnim segmentima eksploatacije motornog vozila, kao što su pouzdanost i ekonomičnost.

Pouzdanost motornog vozila se umanjuje pojavom bilo kakve, makar i najsitnije neispravnosti na njegovom pogonskom agregatu. Osim toga, sitna neispravnost može vremenom da preraste u krupniju neispravnost, koja će u jednom trenutku rezultirati otkazom nekog sistema motora, a to može ugroziti bezbednost putnika u konkretnom vozilu, ali i bezbednost ostalih učesnika u saobraćaju.

Ekonomski aspekt eksploatacije motornih vozila takođe je pogođen pojavom pomenutih sitnijih neispravnosti na motoru, jer one u najvećem broju slučajeva dovode do povećane potrošnje goriva. Osim toga, kako je malopre rečeno, dugotrajan rad motora u neregularnim režimima (u prisustvu neispravnosti) može da dovede do otkaza, a to takođe iziskuje dodatne troškove za vlasnika vozila, koji mora da plati popravku.

Iz svih navedenih razloga, proizilazi da male neispravnosti mogu da dovedu do velikih problema, tim pre, što od momenta pojave jedne takve neispravnosti pa do njenog detektovanja i identifikovanja, može da prođe dosta vremena. Zbog toga se dijagnostikovanje ovakvih neispravnosti smatra slabom tačkom današnjih OBD sistema, pa istraživači koji se bave ovom oblašću ulažu velike napore, kako bi pronašli nove metode i

tehnike za prevazilaženje tog problema. To čini ovu problematiku veoma aktuelnom, pa je u skladu sa time odabran i predmet istraživanja u ovoj disertaciji.

U odnosu na veoma obiman problem dijagnostike na motornim vozilima, koji svakim danom postaje sve obimniji, jer se povećava broj senzora čije ponašanje treba pratiti, u ovom radu će fokus biti na dijagnostici usisnog sistema oto motora. Predmet istraživanja ovog rada su otkrivanje i prepoznavanje manjih neispravnosti u usisnom sistemu koje mogu dovesti do prekomernog pogoršanja izduvne emisije motora ili nekih drugih neželjenih posledica. Imajući u vidu ranije dato obrazloženje u vezi sa sensorima, odabrane su neispravnosti pogrešnog očitavanja masenog protoka vazduha, kao i pritiska i temperature u usisnom kolektoru, koje mogu da utiču na sastav smeše. Osim toga, predmet istraživanja u ovom radu je i otkrivanje nepredviđenog ulaska vazduha u usisni sistem (u žargonu: „falš“-vazduh), jer ni ovu pojavu nije lako otkriti, s obzirom da je u pitanju mehanička neispravnost. Do pojave „falš“-vazduha može doći iz raznih razloga, kao što su oštećenja u usisnom sistemu, oštećenja na crevima koja povezuju usisni sistem sa drugim sistemima na vozilu (npr. crevo sa servo uređaja kočnice), labav spoj creva sa usisnim sistemom i sl. Pored neispravnosti senzora i „falš“-vazduha, uzeta je u razmatranje još jedna neispravnost, koja ima uticaja na emisiju izduvnih gasova. To je umanjen protok recirkulisanih izduvnih gasova, koji ulaze u usisni sistem, što može biti posledica zagušenja u EGR vodu usled gareži i drugih nečistoća, ili zaglavljenog EGR ventila.

## 1.2 Cilj istraživanja i hipoteza

Glavni cilj istraživanja je bio da se prouči ponašanje automobilske oto motora tokom rada na ustaljenom režimu, kada u usisnom sistemu tog motora postoji neka neispravnost i da se na osnovu toga razvije jedan dijagnostički model usisnog sistema, koji bi bio zasnovan na eksperimentalno dobijenim podacima. Ovaj model bi trebalo da posluži za otkrivanje i prepoznavanje predefinisanih tipova neispravnosti u usisnom sistemu motora tokom rada na ustaljenom režimu, za različite intenzitete neispravnosti. Kako je navedeno u opisu problema istraživanja, radi se o sledećim tipovima neispravnosti: nepredviđeni ulazak vazduha u usisni sistem („falš“-vazduh), pogrešno očitavanje davača pritiska i temperature u usisnom kolektoru, pogrešno očitavanje davača masenog protoka vazduha i umanjeni protok recirkulisanih izduvnih gasova. Međutim, izazivanje ovih neispravnosti na stvarnom motoru može da bude veoma štetno po sâm motor, naročito kada se uzme u obzir veliki broj takvih eksperimenata koje treba izvesti tokom istraživanja, kako bi se obuhvatile različite veličine tih neispravnosti u dovoljno širokom opsegu rada motora. Tako se ukazala potreba za razvojem jednog simulatora neispravnosti, tj. matematičkog modela motora, pomoću kojeg bi se mogle simulirati različite neispravnosti u njegovom usisnom sistemu. Ova potreba je dovela do definisanja još jednog cilja istraživanja: izraditi softverski simulator neispravnosti, koji će omogućiti izvođenje neograničenog broja virtuelnih eksperimenata sa simuliranjem posmatranih tipova neispravnosti u različitim intenzitetima. Realizacija ovog cilja istraživanja je, hronološki posmatrano, ispred glavnog cilja, definisanog na početku ovog odeljka, jer se razvoj dijagnostičkog modela usisnog sistema zasniva na podacima koje generiše simulator neispravnosti.

Iz ovako definisanih ciljeva istraživanja, proizilazi i hipoteza, koju bi ovaj rad trebalo da potvrdi ili opovrgne. Njena formulacija glasi:

*„Moguće je razviti dijagnostički model usisnog sistema automobilske oto motora, koji će u najvećem broju slučajeva otkriti i prepoznati predefinisane tipove neispravnosti u usisnom sistemu, tokom rada motora na ustaljenom režimu.“*

## 1.3 Struktura disertacije

Disertacija je organizovana po poglavljima koja su pobrojana u sledećoj tabeli:

Tab. 1.1 Struktura disertacije

Poglavlje 1 UVOD
Poglavlje 2 PREGLED STANJA U OBLASTI DIJAGNOSTIKE NEISPRAVNOSTI USISNOG SISTEMA MOTORA
Poglavlje 3 PRIMENA VEŠTAČKIH NEURONSKIH MREŽA U PREPOZNAVANJU UZORAKA
Poglavlje 4 OSREDNJENI MODEL MOTORA
Poglavlje 5 SIMULATOR NEISPRAVNOSTI U USISNOM SISTEMU
Poglavlje 6 DIJAGNOSTIČKI MODEL BAZIRAN NA REZULTATIMA SIMULACIJA
Poglavlje 7 DIJAGNOSTIČKI MODEL ZASNOVAN NA LABORATORIJSKI DOBIJENIM PODACIMA
Poglavlje 8 ZAKLJUČAK I PRAVCI DALJIH ISTRAŽIVANJA
Poglavlje 9 LITERATURA
Poglavlje 10 PRILOZI

U **Poglavlju 1** opisan je predmet istraživanja disertacije. Nakon toga su definisani ciljevi istraživanja, na osnovu kojih je postavljena hipoteza. Potom je prikazana struktura rada, a na kraju su istaknuti glavni naučni doprinosi disertacije.

**Poglavlje 2** prikazuje presek stanja u oblasti dijagnostikovanja neispravnosti u usisnom sistemu motora sa unutrašnjim sagorevanjem. Na početku su date definicije osnovnih pojmova, kao i opis principa, na kojima se zasnivaju metode detektovanja i dijagnostikovanja neispravnosti automatski upravljanih tehničkih sistema uopšte. Posle toga su predstavljene metode detektovanja i dijagnostikovanja neispravnosti, koje se najčešće koriste u istraživanjima vezanim za automobilske motore. Na kraju su dati kratki prikazi određenog broja značajnijih radova koji su se bavili detektovanjem i dijagnostikovanjem neispravnosti u usisnom sistemu motora SUS.

**Poglavlje 3** opisuje osnovna svojstva veštačkih neuronskih mreža sa naglaskom na nadgledane neuronske mreže bez povratnih sprega. Zatim se daju matematičke osnove ovih neuronskih mreža i prikaz procesa njihove obuke pomoću najčešće korišćenog algoritma sa prostiranjem greške unazad. Na kraju ovog poglavlja dat je detaljan prikaz kompletnog postupka razvoja veštačke neuronske mreže sa aspekta prepoznavanja uzoraka i njihovog razvrstavanja u neku od unapred definisanih klasa, jer ta oblast primene neuronskih mreža najpribližnije odgovara rešavanju dijagnostičkih problema.

**Poglavlje 4** upoznaće čitaoca sa osrednjenim modelima motora sa unutrašnjim sagorevanjem, nakon čega sledi diskusija o poziciji ove vrste modela u globalnom skupu modela koji se koriste u oblasti motora SUS. Zatim sledi detaljan prikaz jednog konkretnog osrednjenog modela motora SUS (Hendriksov osrednjeni model motora), koji je kasnije



iskorišćen kao osnova za razvoj softverskog simulatora neispravnosti u okviru ove disertacije.

Podroban prikaz strukture pomenutog simulatora neispravnosti donosi **Poglavlje 5**. Tamo se daje i kompletna metodologija koja opisuje kako je razvijen taj model za simuliranje neispravnosti, tj. kako je modelovan svaki tip neispravnosti ponaosob i to u otvorenoj i zatvorenoj sprezi sa lambda-sondom, za slučajeve motora sa i bez masenog protokomera vazduha. U okviru ovog poglavlja dat je i detaljan opis algoritma računarskog programa, u kojem je implementiran model simulatora neispravnosti. Poglavlje 5 se završava prikazom rezultata primene razvijenog simulatora neispravnosti u svim karakterističnim slučajevima, i detaljnim tumačenjima dobijenih rezultata.

**Poglavlje 6** donosi opis koncepta dijagnostičkog modela, baziranog na podacima koji su dobijeni pomoću simulatora neispravnosti. Osim toga, prikazana je i kompletna metodologija razvoja konkretnih neuronskih mreža koje ulaze u sastav pomenutog dijagnostičkog modela, sa pseudokodovima odgovarajućih računarskih programa, namenski napisanim za tu svrhu. Na kraju poglavlja data je selekcija rezultata testiranja dijagnostičkog modela, sa odgovarajućom diskusijom.

**Poglavlje 7** obuhvata prikaz eksperimentalne verifikacije predloženog dijagnostičkog koncepta. Do detalja je opisano eksperimentalno postrojenje, koje je pripremljeno i upotrebljeno za prikupljanje podataka neophodnih za razvoj laboratorijskog dijagnostičkog modela usisnog sistema. Potom je dat prikaz plana izvođenja eksperimenata, kao i opis njihove realizacije. Zatim slede rezultati dobijeni eksperimentalnim merenjima, a onda i metodologija razvoja dijagnostičkog modela nad prikupljenim podacima. Završetak poglavlja je rezervisan za rezultate testiranja modela i odgovarajuće komentare.

U **Poglavlju 8** daju se zaključni komentari i mogući pravci u kojima bi se odvijao nastavak istraživanja na ovu temu.

Lista korišćenih referenci data je u **Poglavlju 9**.

Na kraju, u **Poglavlju 10** su dati **prilozi**, koji obuhvataju prikaz Simulink modela simulatora neispravnosti, kao i izvorne kodove svih programa koji su napisani u cilju realizacije ovog istraživanja.

## 1.4 Značaj i doprinos disertacije

Naučni doprinos i značaj ove disertacije ogledaju se u sledećem:

1. Proširena je funkcionalnost Hendricks-ovog osrednjenog modela motora sa više aspekata:
  - 1.1 U izvorni Hendricks-ov model integrisane su opšta jednačina uzdužne dinamike vozila i prateće jednačine za određivanje otpora kretanja vozila.

Ovom modifikacijom omogućeno je simuliranje uticaja različitih parametara, od kojih zavise otpori kretanja vozila, na izlazne veličine Hendricks-ovog modela motora. Konkretno, primenom poboljšanog Hendricks-ovog modela mogu se kvalitativno i kvantitativno istraživati uticaji uzdužnog nagiba puta, stepena prenosa u kojem se vozilo kreće, brzine vetra, mase vozila itd. na broj obrtaja motora, pritisak u usisnom kolektoru i temperaturu u sisnom kolektoru.
  - 1.2 Izvedena je jednačina za modelovanje masenog protoka vazduha, koji nepredviđeno ulazi u usisni sistem motora, a potom su u Hendricks-ovom modelu, u skladu sa time, modifikovane jednačine promene pritiska i

temperature u usisnom kolektoru, kao i jednačina za modelovanje potrebnog masenog protoka goriva.

Ovako izvršena nadgradnja Hendricks-ovog modela omogućava simuliranje ustrojavanja neželjenog vazduha u usisni sistem motora i proučavanje uticaja ove neispravnosti na broj obrtaja motora, pritisak u usisnom kolektoru i temperaturu u usisnom kolektoru.

- 1.3 U izvorni Hendricks-ov model integrisane su jednačine za modelovanje potrebnog masenog protoka goriva, kada u usisnom sistemu postoje sledeće neispravnosti: pogrešno očitavanje senzora masenog protoka vazduha, pogrešno očitavanje senzora pritiska u usisnom kolektoru, pogrešno očitavanje senzora temperature u usisnom kolektoru i umanjeni protok recirkulisanih izduvni gasova.

Ova novina u izvornom Hendricks-ovom modelu omogućava simuliranje navedenih neispravnosti u usisnom sistemu motora i proučavanje uticaja tih neispravnosti na broj obrtaja motora, pritisak u usisnom kolektoru i temperaturu u usisnom kolektoru.

- 1.4 Jednačine za modelovanje potrebnog masenog protoka goriva u unapređenom Hendricks-ovom modelu prilagođene su i motorima kod kojih se maseni protok vazduha meri i motorima kod kojih se maseni protok vazduha računa.

Ovim poboljšanjem dobija se na univerzalnosti, tj. omogućava se korišćenje modifikovanog modela za simuliranje neispravnosti u usisnom sistemu motora koji poseduju maseni protokomer vazduha, ali i onih koji ga ne poseduju.

- 1.5 Uvedene su jednačine za korigovanje potrebnog masenog protoka goriva u skladu sa željenom vrednošću koeficijenta viška vazduha, kada u usisnom sistemu postoji neka neispravnost.

Ove jednačine omogućavaju ispitivanje uticaja razmatranih neispravnosti na broj obrtaja motora, pritisak u usisnom kolektoru i temperaturu u usisnom kolektoru, pri radu motora u zatvorenoj sprezi sa lambda-sondom.

2. Izrađen je računarski program koji, u sprezi sa unapređenim Hendriksovim modelom, omogućava simuliranje različitih intenziteta razmatranih tipova neispravnosti.
3. Dati su detaljna analiza i tumačenje uticaja svih simuliranih neispravnosti na pritisak i temperaturu u usisnom kolektoru, kao i na broj obrtaja motora, za slučajeve motora sa masenim protokomerom i bez njega, pri radu u otvorenoj i zatvorenoj sprezi sa lambda sondom.
4. Predložen je novi dijagnostički koncept usisnog sistema motora, u okviru kojeg je razvijen simulacioni dijagnostički model, baziran na neuronskim mrežama, za otkrivanje i prepoznavanje neispravnosti pri radu motora na ustaljenom režimu rada, u otvorenoj i zatvorenoj sprezi sa lambda-sondom.
5. Laboratorijskim eksperimentima je ispitan, a potom i detaljno analiziran, uticaj neispravnosti tipa „ulaz neželjenog vazduha u usisni sistem“ i „pogrešno očitavanje senzora pritiska u usisnom kolektoru“ na broj obrtaja motora, pritisak u usisnom

kolektoru i temperaturu u usisnom kolektoru. Ova laboratorijska ispitivanja sprovedena su na motoru bez protokomera vazduha, koji je radio na ustaljenom režimu, u zatvorenoj sprezi sa lambda-sondom.

6. Na osnovu podataka, dobijenih laboratorijskim eksperimentima, razvijen je dijagnostički model usisnog sistema stvarnog motora, po uzoru na simulacioni dijagnostički model.
7. Razvijen je i implementiran računarski alat koji omogućava:
  - automatsko generisanje i testiranje neuronskih mreža za detektovanje i identifikovanje neispravnosti u usisnom sistemu i
  - automatsko generisanje vizuelnog prikaza dobijenih dijagnostičkih rezultata.

## 2 PREGLED STANJA U OBLASTI DIJAGNOSTIKE NEISPRAVNOSTI U USISNOM SISTEMU MOTORA

### 2.1 Terminologija

S obzirom da je dijagnostika neispravnosti prisutna u svakoj oblasti tehnike, postoji problem neusaglašenosti dijagnostičkih pojmova u pojedinim oblastima. Bilo je više pokušaja da se taj problem prevaziđe i da se ovi pojmovi na neki način standardizuju. U ovom radu biće korišćeni termini, preuzeti iz [2], koje je ustanovio IFAC<sup>2</sup>-ov Tehnički komitet SAFEPROCESS, a koji se inače najviše koriste među istraživačima u ovoj oblasti. No, bez obzira na pomenutu unifikaciju dijagnostičkih pojmova prema IFAC-u, problem terminologije još uvek postoji, naročito kada se posmatra iz perspektive srpskog jezika. Naime, autor ovog rada nije pronašao standardne nazive na srpskom jeziku za sve dijagnostičke pojmove sa engleskog jezika, pa su u tim slučajevima korišćeni sopstveni slobodni prevodi. U nastavku su data tumačenja nekoliko najvažnijih dijagnostičkih pojmova, od kojih su neki u veoma tesnoj vezi sa temom istraživanja ovog rada.

**Neispravnost** (eng. *fault*) – nedozvoljeno ili neželjeno odstupanje bar jednog karakterističnog svojstva (karakteristike) sistema u odnosu na uobičajeno stanje ispravnog sistema. To je jedno neželjeno stanje, koje može dovesti do umanjenja ili gubitka sposobnosti nekog dela tehničkog sistema da izvršava zahtevani zadatak. Postojanje neispravnosti je često teško otkriti, naročito ako je ona mala ili skrivena. Neispravnost može inicirati pojavu otkaza ili privremenog zastoja u radu nekog tehničkog sistema.

**Otkaz** (eng. *failure*) – trajan gubitak sposobnosti sistema da izvršava zahtevanu funkciju u definisanim radnim uslovima. Drugim rečima, otkaz je događaj, koji dovodi do prekida rada sistema. Do otkaza obično dolazi posle početka rada sistema ili pri povećanju opterećenja sistema. Otkaz može da nastane kao rezultat jedne ili više neispravnosti.

**Zastoj u radu** (eng. *malfunction*) – privremena nepravilnost u izvršavanju željene funkcije sistema. Drugim rečima, to je privremeni prekid funkcionisanja sistema. Slično otkazu, i zastoj u radu je događaj, koji nastaje iz jedne ili više neispravnosti, ali posle nekog vremena sistem ponovo funkcioniše, za razliku od otkaza.

Odnos između neispravnosti, otkaza i zastoja u radu ilustrovan je na slici *Sl. 2.1*. Na toj slici vidi se da otkaz ili zastoj u radu mogu nastati kao posledica neispravnosti. Takođe, vidi se i da se promena relevantne karakteristike sistema može razvijati postepeno ali može biti i nagla. Izlazak relevantne karakteristike izvan granica tolerancije za normalne vrednosti ukazuje na pojavu neispravnosti u trenutku  $t_1$ . U zavisnosti od vrste i veličine neispravnosti, u nekom trenutku  $t_2$  može doći do otkaza ili zastoja u radu sistema.

**Greška** (eng. *error*) – odstupanje izmerene ili izračunate vrednosti neke veličine u odnosu na njenu tačnu vrednost

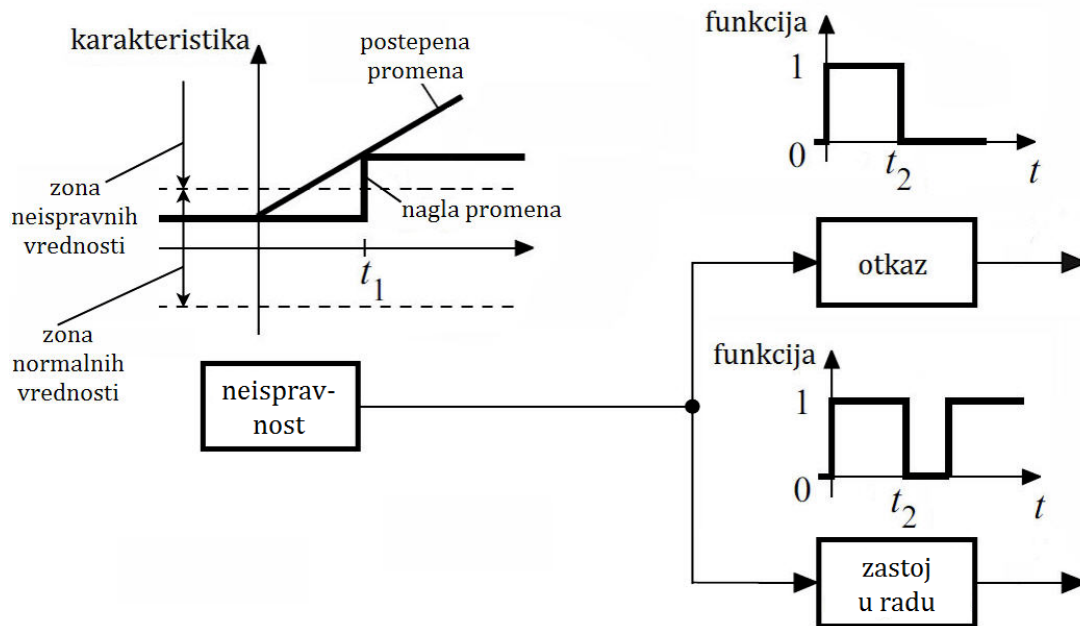
**Poremećaj** (eng. *disturbance*) – nepoznati (i nekontrolisani) ulaz koji deluje na sistem.

**Rezidual** (eng. *residual*) – indikator neispravnosti, zasnovan na odstupanju izmerene vrednosti neke veličine od njene izračunate vrednosti

**Simptom** (eng. *symptom*) – promena neke osmotrive (opservabilne) veličine u odnosu na njeno normalno ponašanje

---

<sup>2</sup> IFAC – International Federation of Automatic Control (Međunarodna federacija za automatsko upravljanje)



Sl. 2.1 Razvoj otkaza i zastoja u radu iz neispravnosti koja uzrokuje postepenu ili iznenadnu promenu nekog karakterističnog svojstva sistema [2]

S obzirom da je predmet interesovanja ovog rada neispravnost, u nastavku se daju definicije još nekoliko pojmova koji su sa njome direktno povezani.

**Detektovanje neispravnosti** (eng. *fault detection*) – određivanje prisustva neispravnosti u sistemu i vremena pojave neispravnosti

**Lokalizovanje neispravnosti** (eng. *fault isolation*) – određivanje vrste, lokacije i vremena pojave neispravnosti procenjivanjem simptoma; sledi posle detektovanja neispravnosti

**Identifikacija neispravnosti** (eng. *fault identification*) – određivanje veličine neispravnosti i njenog ponašanja u vremenu; sledi posle lokalizovanja neispravnosti

**Dijagnostikovanje neispravnosti** (eng. *fault diagnosis*) – određivanje vrste, veličine, lokacije i vremena detektovanja neispravnosti procenjivanjem simptoma. Sledi posle detektovanja neispravnosti.

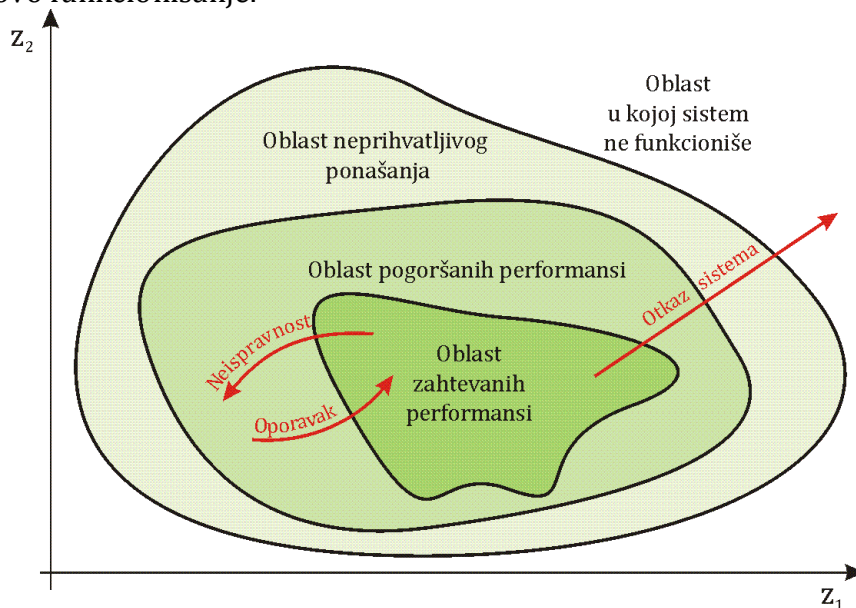
S obzirom da je prema IFAC-ovoj terminologiji pojam detektovanja neispravnosti definisan kao zaseban, tj. nije obuhvaćen pojmom dijagnostikovanja neispravnosti, u radovima koji se bave ovom oblašću često se pominju i pojmovi, obeleženi skraćenicama FDD (Fault Detection and Diagnosis) i FDI (Fault Detection and Isolation). Razlog zbog kojeg su uvedeni ovi složeni pojmovi je taj, što se pri istraživanjima vezanim za lokalizovanje ili dijagnostikovanje neispravnosti, nikako ne može preskočiti detektovanje neispravnosti.

## 2.2 Uopšte o detektovanju i dijagnostikovanju neispravnosti kod automatski upravljanih sistema

U svakom tehničkom sistemu mogu se pojaviti neispravnosti koje imaju negativan uticaj na njegovo dalje funkcionisanje. Tako, na primer, neispravnosti aktuatora pogoršavaju performanse upravljačkog dela nekog tehničkog sistema, a mogu dovesti i do prestanka funkcionisanja upravljanog sistema. Dalje, pogrešno očitavanje nekog senzora vrlo često je uzrok znatnog odstupanja radne tačke sistema u odnosu na optimalnu radnu tačku. Zatim, pohabanost ili neka druga oštećenja pojedinih komponenti tehničkog sistema umanjuju efikasnost i kvalitet funkcionisanja tehničkog sistema. U mnogim situacijama kada postoji

neispravnost u nekom postrojenju, mora se čak i zaustaviti rad tog postrojenja, kako bi se izbegla njegova oštećenja ili povređivanje prisutnih lica. Iz svih navedenih razloga, detektovanje i dijagnostikovanje neispravnosti, ali i upravljanje tim neispravnostima dobijaju, sve važniju ulogu u modernim tehničkim sistemima sa velikim brojem visoko-automatizovanih i međusobno povezanih komponenti, gde neispravnost jedne komponente može da izbaciti iz stroja ceo sistem.

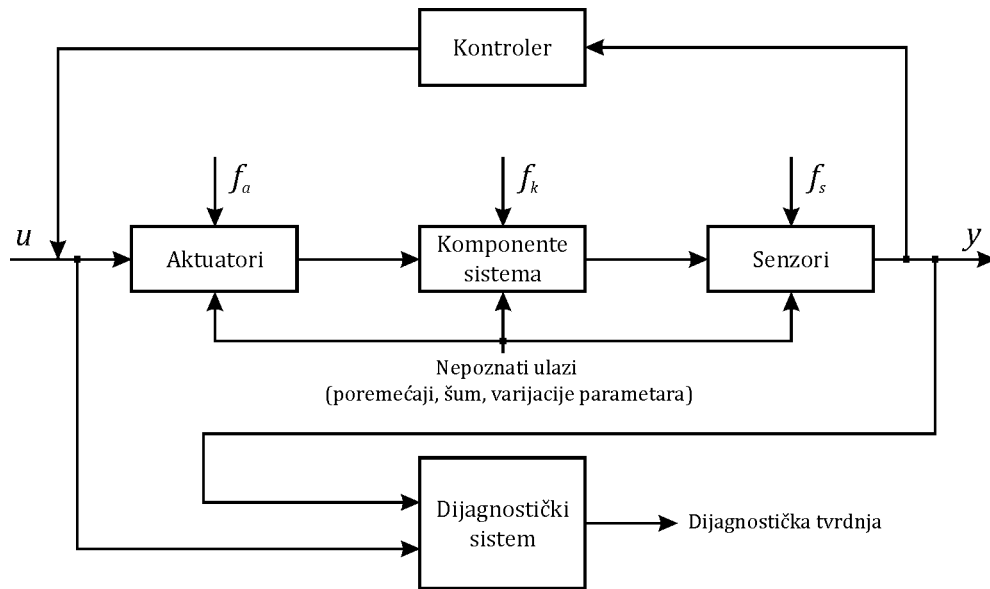
Sa aspekta automatski upravljanih sistema (a automobilski motor je svakako jedan od njih), upravljanje neispravnostima je veoma važno i najčešće podrazumeva dodatne radnje u cilju prilagođavanja upravljačkih algoritama novonastaloj situaciji, kako bi se održalo normalno funkcionisanje sistema u uslovima kada u njemu postoji neka neispravnost. Ovakav način upravljanja sistemima naziva se *upravljanje tolerantno na neispravnosti*<sup>3</sup> i veoma je aktuelan u poslednje vreme. Da bi se ovakvo upravljanje moglo realizovati, neophodno je ustanoviti da u postrojenju postoji neka neispravnost (što predstavlja detektovanje neispravnosti), a zatim i odrediti na kojoj se komponenti postrojenja ta neispravnost pojavila, kao i kolika je veličina neispravnosti (što pripada domenu dijagnostike neispravnosti). Ove informacije su veoma važne, jer na osnovu njih, upravljački sistem postrojenja vrši korekciju parametara u upravljačkom algoritmu obezbeđujući nesmetano funkcionisanje postrojenja, bez obzira na prisustvo neispravnosti. Na taj način, detektovanje i dijagnostikovanje neispravnosti u sistemima, koji imaju *upravljanje tolerantno na neispravnosti* još više dobijaju na značaju. Ovo je ilustrovano na Sl. 2.2, gde su prikazane oblasti funkcionisanja jednog takvog sistema, pri čemu je pretpostavljeno da se njegove performanse mogu opisati sa neke dve promenljive,  $z_1$  i  $z_2$ . Uprkos raznim poremećajima ili čak i nekim manjim neispravnostima, koje prevode posmatrani sistem iz oblasti zahtevanih performansi u oblast pogoršanih performansi, upravljanje treba da ima sposobnost da promenljive  $z_1$  i  $z_2$  vrati u granice, pri kojima će posmatrani sistem ponovo imati zahtevane performanse. U suprotnom, posmatrani sistem će funkcionisati sa pogoršanim performansama. Ako postoje veće neispravnosti, one dovode do neprihvatljivog ponašanja, a u slučaju izuzetno velikih neispravnosti, ili dolazi do otkaza posmatranog sistema ili se aktivira sigurnosni sistem (ako postoji) koji zaustavlja njegovo funkcionisanje.



Sl. 2.2 Oblasti funkcionisanja postrojenja sa upravljanjem tolerantnim na neispravnosti [3]

<sup>3</sup> eng. Fault-Tolerant Control

Na Sl. 2.3 prikazana je uprošćena šema jednog automatski upravljanog sistema, čiji su osnovni podsistemi aktuatori, komponente i senzori. Na osnovu upravljačkih signala  $u$ , dobijenih od kontrolera, aktuatori izvedu određena dejstva nad komponentama sistema. Usled ovih dejstava menjaju se vrednosti veličina koje su relevantne za pojedine komponente. Zatim senzori mere ove vrednosti i odgovarajuće merne signale  $y$  šalju kontroleru. Neispravnosti se mogu pojaviti i na aktuatorima ( $f_a$ ) i na komponentama sistema ( $f_k$ ) i na sensorima ( $f_s$ ). U slučaju automobilskog motora, kao primer neispravnosti aktuatora može se navesti nedovoljno podizanje igle brizgača goriva, kao neispravnost komponente tretira se, na primer, isticanje goriva iz nekog creva, a jedna od neispravnosti senzora je npr. pogrešno očitavanje protoka vazduha.



Sl. 2.3 Šema automatski upravljanog (nadgledanog) sistema

Neispravnosti aktuatora, komponenti sistema i senzora opisuju se kao ulazi u sistem, i njihovo postojanje doprinosi promeni vrednosti izlaznog vektora  $y$ . Osim neispravnosti  $f_a$ ,  $f_k$  i  $f_s$ , tu su još i nepredviđeni ulazi u sistem u vidu raznih poremećaja, mernog šuma, varijacija parametara zbog starenja posmatranog sistema tokom čitavog perioda eksploatacije itd. Ovi nepoznati ulazi najčešće nemaju neki značajan uticaj na samo funkcionisanje posmatranog sistema, ali ipak na neki način utiču na njegovo ponašanje, čime otežavaju procese detektovanja i dijagnostikovanja neispravnosti  $f_a$ ,  $f_k$  i  $f_s$ .

Vektor upravljačkih signala  $u$  i vektor mernih signala  $y$  predstavljaju ulazne veličine dijagnostičkog sistema. Na osnovu ovih vektora dijagnostički sistem u realnom vremenu daje rezultat u vidu dijagnostičke tvrdnje. Ova dijagnostička tvrdnja sadrži odgovor na pitanje da li je u nadgledanom sistemu trenutno prisutna neka neispravnost, kao i informacije o lokaciji i veličini eventualne neispravnosti. Da bi dijagnostički sistem generisao ispravnu dijagnostičku tvrdnju, on mora da sadrži neko znanje o očekivanom (normalnom) ponašanju nadgledanog sistema, kako bi imao sa čime da uporedi njegovo trenutno ponašanje, opisano vektorima  $u$  i  $y$ . U zavisnosti od toga, u kom obliku je znanje o normalnom ponašanju nadgledanog sistema predstavljeno dijagnostičkom sistemu, razvili su se različiti pristupi u detektovanju i dijagnostikovanju neispravnosti.

## 2.3 Pristupi u detektovanju i dijagnostikovanju neispravnosti

Postoji veliki broj različitih metoda koje se koriste za detektovanje i dijagnostikovanje neispravnosti u tehničkim sistemima. Radi lakšeg izučavanja, ove metode su u svetskoj

literaturi razvrstavane u određene grupe. Postoji više radova, u kojima su autori pokušavali da naprave neku sistematizaciju tih metoda. Međutim, problem je što se povremeno pojavljuju nove metode koje predstavljaju kombinacije metoda poznatih od ranije i zbog toga se ne uklapaju u postojeće sistematizacije u potpunosti. Zbog velikog broja metoda, ovde će biti data veoma kratka objašnjenja nekih karakterističnih, a detaljnije informacije o njihovom istorijatu i sistematizaciji mogu se pronaći u publikacijama [4], [5], [6], [7], [8], [9] i [10].

Prema Gertleru [4], sve metode detektovanja i dijagnostikovanja neispravnosti u nekom tehničkom sistemu zasnivaju se na jednom od dva osnovna pristupa:

- Tradicionalni (nemodelski) pristup i
- Modelski pristup

U metodama tradicionalnog pristupa, ne koriste se nikakvi modeli, da bi se dijagnostičkom sistemu predstavilo znanje o normalnom ponašanju nadgledanog sistema ili da bi se došlo do dijagnostičke tvrdnje.

Sa druge strane, kod metoda modelskog pristupa uvek je prisutan neki model koji opisuje ili pojedine procese u tehničkom sistemu ili same neispravnosti ili kompletan tehnički sistem. U tom slučaju, takvi modeli su deo dijagnostičkog sistema i oni su nosioci znanja o nadgledanom sistemu.

U nastavku će ukratko biti navedene osnovne specifičnosti po nekoliko najvažnijih metoda iz oba pomenuta pristupa.

### 2.3.1 Tradicionalni (nemodelski) pristup

Od tradicionalnih metoda najstarija i najjednostavnija je *metoda proverе granica*. Ona se zasniva na poređenju izmerenih vrednosti nekih relevantnih veličina posmatranog sistema sa unapred zadatim graničnim vrednostima. Kod ove metode, znanje o očekivanom ponašanju nadgledanog sistema sadržano je upravo u tim graničnim vrednostima, koje ne moraju biti konstantne tokom funkcionisanja sistema, jer se mogu menjati u zavisnosti od njegove radne tačke. Međutim, bez obzira na to, da li su granične vrednosti konstantne ili promenljive u vremenu, ukoliko izmerena vrednost neke veličine nije unutar zadatih granica, to ukazuje na postojanje neispravnosti u sistemu. Osim jednostavnosti i niske cene implementiranja ove metode, postoje i neki nedostaci. Glavni nedostatak je ograničena uspešnost detektovanja neispravnosti, jer mnoge neispravnosti uopšte ne moraju da uzrokuju izlazak vrednosti karakterističnih veličina izvan dozvoljenog opsega. Dalje, pomoću ove metode se ne može izvršiti lokalizacija neispravnosti, osim u izuzetnim slučajevima ili u sadejstvu sa nekom drugom metodom.

*Metoda korišćenja posebnih senzora* podrazumeva instalisanje specijalnih senzora koji ili reaguju na granične dozvoljene vrednosti merenih veličina (što se na neki način preklapa sa prvom nemodelskom metodom) ili mere vrednosti nekih posebnih veličina kao što su vibracije ili jačina zvuka koje emituje posmatrani tehnički sistem.

*Metoda hardverske ili fizičke redundantnosti* koristi se uglavnom za detektovanje ili lokalizovanje neispravnosti senzora. Ova metoda podrazumeva instaliranje više senzora koji mere vrednost najčešće neke kritične fizičke veličine u sistemu. Bilo kakvo ozbiljnije neslaganje između izmerenih vrednosti ukazuje na neispravnost nekog od pomenutih senzora. Ukoliko se koriste dva takva senzora, može se samo detektovati neispravnost jednog od njih. Međutim, ukoliko su instalirana tri senzora, onda se pomoću ove metode može izvršiti i lokalizacija neispravnosti senzora, tj. može se utvrditi koji od njih je neispravan. Kod ove metode, znanje o normalnom ponašanju nadgledanog sistema (u ovom slučaju senzora) sadržano je u vrednostima koje očitavaju ispravni senzori.

Jasno je da metoda fizičke redundantnosti ima značajan nedostatak u vidu povećanja cene postrojenja, zbog ugradnje više senzora. Osim ekonomskih razloga, nedostatak



prostora za ugradnju dodatnih senzora takođe može biti važan razlog koji ograničava masovniju primenu ove metode. Ipak, velika pouzdanost ove metode doprinela je da se ona uglavnom primenjuje u sistemima u kojima pojava neispravnosti nekog kritičnog senzora može da ima katastrofalne posledice. Takvi sistemi su, na primer, nuklearna postrojenja, letelice i slično.

*Metoda frekventne analize* mernih signala zasniva se na analizi frekventnog spektra karakterističnih merenih veličina u posmatranom sistemu. Pri tome se koristi činjenica da je poznat frekventni spektar posmatranih mernih signala pri normalnom funkcionisanju sistema. Svako odstupanje od tog spektra ukazuje da se pojavila neka neispravnost u sistemu. Osim detektovanja, pomoću ove metode je moguće i lokalizovati neku neispravnost, ukoliko je poznat karakteristični frekventni spektar za tu konkretnu neispravnost.

Nemodelske metode se uglavnom koriste u slučajevima kada je posmatrani tehnički sistem toliko komplikovan da je nemoguće napraviti model sistema koji će biti zadovoljavajućeg kvaliteta sa aspekta detektovanja i dijagnostikovanja neispravnosti. U suštini, nemodelske metode su ili nedovoljno uspešne u detektovanju i dijagnostikovanju neispravnosti ili, ako su uspešne (kao što je slučaj sa metodom fizičke redundantnosti), onda cena njihove primene značajno raste. Zbog toga je fokus istraživanja u oblasti detektovanja i dijagnostikovanja neispravnosti u tehničkim sistemima u poslednjih dvadesetak godina usmeren uglavnom na metode modelskog pristupa.

### 2.3.2 Modelski pristup

Kako je već rečeno, osnovna karakteristika modelskog pristupa je postojanje neke vrste modela, koji opisuje relevantne procese u posmatranom tehničkom sistemu. Ovi modeli se mogu zasnivati na različitim principima, pa, u skladu sa time, postoje i različite grupe metoda za detektovanje i dijagnostikovanje neispravnosti u tehničkim sistemima. U referencama [5], [6], [7] i [8] dati su pregledi velikog broja metoda zasnovanih na modelskom pristupu i na osnovu toga, sve modelske metode za detektovanje i dijagnostikovanje neispravnosti u tehničkim sistemima mogle bi se podeliti na:

- kvalitativne i
- kvantitativne.

U nastavku će ukratko biti opisane kvalitativne i kvantitativne metode čija je primena moguća u dijagnostičkim sistemima motornih vozila, ali treba istaći da su u okviru jednog dijagnostičkog sistema vrlo često prisutne različite kombinacije kvalitativnih i kvantitativnih metoda.

**Kvalitativne modelske metode** podrazumevaju primenu simboličkog znanja u modelovanju tehničkog sistema koji je predmet detektovanja i dijagnostikovanja neispravnosti. Ove metode uglavnom pripadaju oblasti veštačke inteligencije, a najčešće među njima su ekspertske sistemi i fazi-logika.

*Ekspertske sistem* u nekoj oblasti je model koji oponaša kognitivno ponašanje stručnjaka pri rešavanju problema iz te oblasti. Osnovne komponente ekspertske sistema su baza znanja i mehanizam zaključivanja. Baza znanja, u suštini, predstavlja skup pravila, najčešće oblika „AKO simptomi ONDA neispravnost“, pomoću kojih se opisuje ponašanje posmatranog sistema na osnovu znanja eksperta o tom sistemu. Mehanizam zaključivanja je algoritam koji pretražuje bazu znanja kako bi izvukao dijagnostički zaključak na osnovu trenutnih podataka o posmatranom sistemu. Međutim, problem kod ekspertske sistema je taj, što se stablo pravila u strukturi baze znanja rapidno uvećava sa povećanjem kompleksnosti posmatranog sistema. Dalje, loša im je sposobnost generalizacije, tj. slabe su im dijagnostičke performanse kada se u posmatranom sistemu pojavi neka situacija koja nije definisana u bazi znanja [7].

Više detalja o ekspertskim sistemima uopšte i njihovoj primeni u detektovanju i dijagnostikovanju neispravnosti može se naći u referencama [11],[12], [13] i [14].

Dijagnostički sistemi na bazi *fazi-logike* uspešno kombinuju teoriju fazi-skupova sa sposobnošću zaključivanja ljudskih bića. Oni su slični ekspertskim sistemima, jer se zasnivaju na pravilima koja su izvedena uglavnom iz ekspertskog znanja o posmatranom tehničkom sistemu. Ova pravila su takođe najčešće oblika „AKO *simptomi* ONDA *neispravnost*“, pri čemu su *simptomi* i *neispravnosti* lingvistički pojmovi koji predstavljaju ulazne i izlazne promenljive dijagnostičkog sistema. Pomenuti lingvistički pojmovi su npr. tipa „potrošnja goriva je mala“ ili „emisija CO je vrlo visoka“, a dobijaju se primenom fazi-logike na konkretne izmerene vrednosti potrošnje goriva, odnosno emisije CO.

Radi ilustracije, u *Tab. 2.1* dato je nekoliko pravila od ukupno 201, koliko ih ima u kompletnoj bazi pravila dijagnostičkog sistema razvijenog u referenci [15].

*Tab. 2.1 Izvod iz baze pravila dijagnostičkog sistema predstavljenog u [15]*

Pravilo br.	Simptomi								Neispravnost
	Snaga motora	Potrošnja goriva	Temperatura motora	Emisija CO	Emisija HC	Emisija CO2	Emisija O2	Koeficijent viška vazduha	
1	Normalna	Normalna	Normalna	Normalna	Normalna	Normalna	Normalna	Normalan	Bez neispravnosti
2	Normalna	Mala	Normalna	Niska	Niska	Niska	Visoka	Visok	Siromašna smeša
3	Mala	Vrlo mala	Vrlo Visoka	Vrlo niska	Vrlo Visoka	Vrlo niska	Vrlo Visoka	Vrlo Visok	Vrlo siromašna smeša
4	Velika	Velika	Normalna	Visoka	Visoka	Niska	Niska	Nizak	Bogata smeša
5	Mala	Vrlo velika	Normalna	Vrlo visoka	Vrlo visoka	Vrlo niska	Vrlo niska	Vrlo niska	Vrlo bogata smeša
6	Normalna	Normalna	Normalna	Visoka	Normalna	Niska	Visoka	Normalan	Loša atomizacija goriva
7	Vrlo mala	Normalna	Normalna	Normalna	Vrlo visoka	Niska	Visoka	Normalan	Neispravan sistem paljenja
8	Mala	Normalna	Normalna	Visoka	Vrlo visoka	Niska	Niska	Nizak	Slaba kompresija
9	Mala	Normalna	Visoka	Normalna	Normalna	Normalna	Normalna	Normalan	Preveliko trenje
10	Vrlo mala	Normalna	Vrlo visoka	Normalna	Vrlo niska	Normalna	Niska	Normalan	Pregrevanje
11	Normalna	Velika	Normalna	Normalna	Normalna	Normalna	Normalna	Normalna	Isticanje goriva iz sistema

Iako su naizgled atraktivne, kvalitativne dijagnostičke metode imaju jedan značajan nedostatak. Naime, u mnogim inženjerskim aplikacijama formiranje baze pravila (bez obzira da li je u pitanju dvovalentna ili polivalentna logika) uopšte nije jednostavan zadatak i zahteva obimno ekspertsko znanje o posmatranom sistemu.

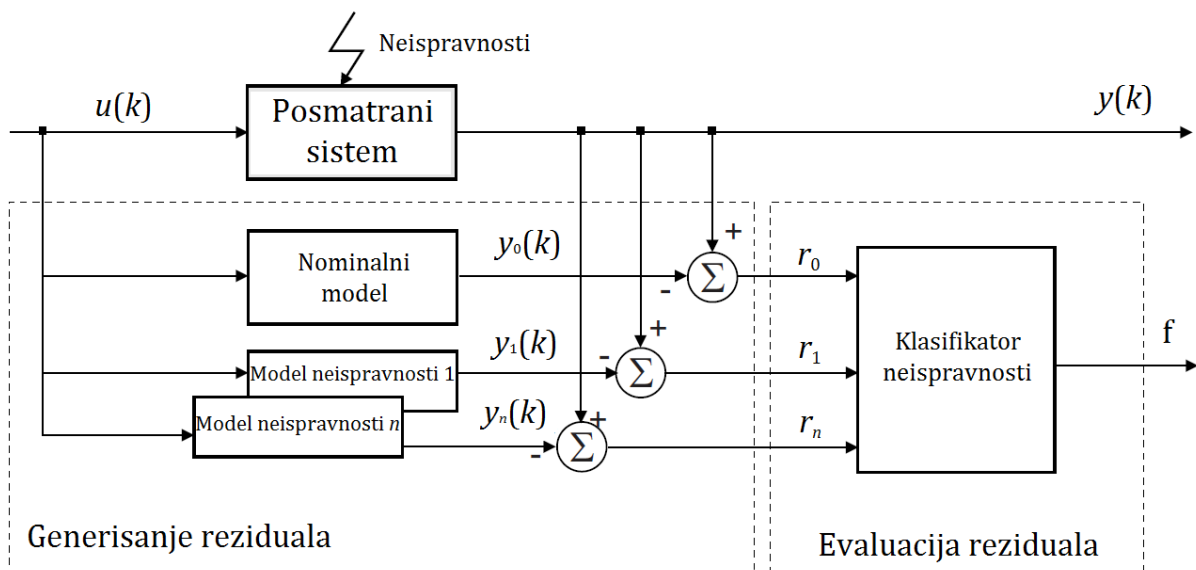
Tačnost dijagnoze u ovom slučaju zavisi od pravila koja se nalaze u bazi, a kreiranje bogate i detaljne baze pravila je zadatak, čija realizacija poprilično dugo traje i zahteva angažovanje većeg broja eksperata.

Iz gore navedenih razloga, kvalitativne metode se u dijagnostičkim sistemima motornih vozila primenjuju uglavnom u kombinaciji sa kvantitativnim metodama.

Osnovna karakteristika **kvantitativnih modelskih metoda** koje se koriste u dijagnostici neispravnosti je realizacija koncepta softverske redundantnosti, kako bi se zamenio uspešan ali skup koncept fizičke (hardverske) redundantnosti [16]. Za razliku od hardverske redundantnosti, ovde je osnovna ideja da se vrednost neke fizičke veličine, koja je izmerena pomoću senzora, poredi sa izračunatom vrednošću iste te veličine. Ta vrednost se računa na osnovu izmerenih vrednosti nekih drugih fizičkih veličina pomoću kvantitativnog modela koji opisuje relacije između njih. Osnovna ideja softverske redundantnosti može se proširiti i na poređenje vrednosti dobijenih samo pomoću modela, pri čemu je svaka od njih izračunata na osnovu različitih mernih i/ili upravljačkih signala.

Bez obzira da li se radi o osnovnoj ili proširenoj ideji softverske redundantnosti, rezultujuće razlike, dobijene poređenjem, nazivaju se reziduali. Dakle, reziduali su veličine koje reprezentuju nekonzistentnost između stvarnog ponašanja sistema i njegovog modela. Iz same prirode reziduala sledi da njegova vrednost treba da bude blizu nule pri normalnom funkcionisanju sistema, a pri pojavi neispravnosti, trebalo bi da dođe do osetnijeg odstupanja od nule.

Generisanje i evaluacija reziduala su ključni procesi koji treba da se sprovedu u svakom dijagnostičkom sistemu koji se zasniva na kvantitativnom modelu. Proces generisanja reziduala zasniva se na poređenju izmerenih i izračunatih vrednosti relevantnih veličina, a proces evaluacije reziduala je posvećen analizi reziduala, da bi se utvrdilo da li postoji neispravnost u posmatranom sistemu i kakva je neispravnost u pitanju. Na Sl. 2.4, ilustrovana je principijelna struktura jednog dijagnostičkog sistema zasnovanog na kvantitativnom modelu, gde se vidi da je za generisanje reziduala upotrebljeno više modela.



Sl. 2.4 Opšta šema modelskog dijagnostičkog sistema [17]

Jedan model (na slici predstavljen kao nominalni) opisuje funkcionisanje ispravnog sistema, a svaki sledeći opisuje prisustvo neke neispravnosti u sistemu. Poređenjem izlaza  $y(k)$  posmatranog sistema u trenutku  $k$  i izlaza modela  $y_0(k)$ ,  $y_1(k)$ , ...,  $y_n(k)$  u istom tom trenutku, dobija se vektor reziduala  $\mathbf{r} = [r_0, r_1, \dots, r_n]$ , koji karakteriše stanje posmatranog sistema sa aspekta ispravnosti. Na kraju, klasifikator neispravnosti izvršava evaluaciju vektora reziduala, tako što ga na neki način transformiše u dijagnostičku tvrdnju  $f$ . Treba istaći da su ovde neophodni i modeli neispravnosti za generisanje reziduala, jer bi, u slučaju samo nominalnog modela, bilo moguće izvršiti detektovanje neispravnosti, ali ne i njeno kompletno dijagnostikovanje.

U procesu generisanja reziduala koriste se egzaktni *matematički (analitički) modeli*, kad god je to moguće. Ovi analitički modeli zasnivaju se na fizičkim zakonitostima i sačinjeni su uglavnom od diferencijalnih jednačina, koje opisuju ponašanje posmatranog tehničkog sistema u ispravnom stanju i pri postojanju određenih neispravnosti. Najpoznatije tehnike koje koriste analitičke modele za generisanje reziduala su: korišćenje opserversa ili estimatora stanja sistema, procena parametara i primena paritetnih jednačina<sup>4</sup>. Detaljnije o ovim tehnikama može se naći u referencama [2], [18] i [19].

<sup>4</sup> eng. Parity equation

Pri korišćenju analitičkih modela za generisanje reziduala, problem je što je veoma teško razviti dovoljno tačan model, naročito ako je u pitanju kompleksan sistem, kao što je automobilski motor. Osim toga, rešavanje većeg broja nelinearnih diferencijalnih jednačina u realnom vremenu takođe može da bude problem za računare kakvi se koriste u automobilima.

Iz navedenih razloga, sve više se za generisanje reziduala koristi veštačka inteligencija, a najčešće *modeli zasnovani na neuronskim mrežama*. Ovakav jedan model zahteva veliku količinu podataka za obuku, da bi „naučio“ kako se ponaša realni posmatrani sistem, ali mu nije potrebno gotovo nikakvo znanje o fizičkim zakonitostima koje određuju ponašanje posmatranog sistema. Podaci za obuku se uglavnom dobijaju eksperimentalnim merenjima ulaznih i izlaznih veličina posmatranog sistema. Prikupljanje podataka za obuku i sâm proces obuke mogu da budu zahtevni sa aspekta utrošenog vremena i računarskih resursa. Međutim, kada je neuronska mreža dobro obučena, ona je sposobna da oponaša bilo koji nelinearni sistem i da prati njegovu dinamiku u realnom vremenu, bez značajnijeg angažovanja računarskih resursa. Ovo je naročito važno kada se uzmu u obzir relativno skromne performanse računara koji upravljaju automobilskim motorima. Za zadatke modelovanja kompleksnih dinamičkih sistema pomoću neuronskih mreža koriste se uglavnom tzv. rekurentne neuronske mreže, a o tome se detaljnije može pročitati u referenci [17].

Proces evaluacije reziduala treba da obezbedi transformaciju vektora reziduala  $r = [r_0, r_1, \dots, r_n]$  u dijagnostičku tvrdnju  $f$  o stanju posmatranog sistema sa aspekta ispravnosti, kako je prikazano na Sl. 2.4. Ovo se može smatrati problemom klasifikacije, gde je cilj da se svaki konkretan uzorak vektora  $r$  svrsta u jednu od unapred definisanih kategorija neispravnosti. U rešavanju ovakvih zadataka najbolje su metode veštačke inteligencije, među kojima se najviše koriste neuronske mreže, fazi neuronske mreže, fazi-logika, stabla odlučivanja i još neke metode zaključivanja, o čemu se više detalja može naći u referencama [18] i [19].

Kao što se može zaključiti iz prethodna dva pasusa, neuronske mreže se mogu koristiti i za generisanje i za evaluaciju reziduala. Razlozi tome su njihove izuzetne sposobnosti aproksimacije veoma složenih funkcija i prepoznavanja uzoraka. Osim toga, otporne su na šum mernih signala i na mernu nesigurnost, što je veoma bitno sa dijagnostičkog aspekta, da bi se minimizovao broj lažnih alarma. Više detalja o neuronskim mrežama, sa akcentom na njihovu primenu u prepoznavanju uzoraka, može se naći u Poglavlju 3.

Naravno, postoje i neki nedostaci primene neuronskih mreža u detektovanju i dijagnostikovanju neispravnosti u tehničkim sistemima. Osim uglavnom dugotrajnog procesa obuke, veliki problem je i otežano dobijanje realnih podataka pri funkcionisanju sistema koji ima neku neispravnost, o čemu će biti malo više reči u Poglavljima 5 i 7.

Dugotrajnim proučavanjem literature koja se bavi dijagnostikom neispravnosti u usisnom sistemu automobilskih motora, autor ovog rada je ustanovio da se napredak u toj oblasti odvija uglavnom razvojem dijagnostičkih metoda baziranih na modelskom pristupu. Pri tome, paralelno se razvijaju egzaktno analitičke metode i metode zasnovane na veštačkoj inteligenciji (pre svega na neuronskim mrežama), a često se u istraživanjima koriste i razne kombinacije pomenutih metoda. Iz tih razloga, u nastavku se daje pregled najvažnijih radova, koji ilustruju razvoj i primenu modelskih dijagnostičkih metoda u istraživanoj oblasti.

## 2.4 Pregled relevantnih radova u oblasti modelskih metoda za dijagnostikovanje neispravnosti u usisnom sistemu motora

### 2.4.1 Radovi koji ilustruju razvoj i primenu analitičkih modela

Rizzoni i saradnici koristili su proširenu verziju linearnog modela prostora stanja motora za detektovanje grešaka senzora položaja leptira i senzora pritiska u usisnom kolektoru i to su opisali u radovima [20] i [21]. Validaciju svog dijagnostičkog modela izvršili su na jednom Fordovom motoru, radne zapremine 3 litra. Pokazalo se da su greške senzora veličine preko 10% uspešno detektovane. Međutim, radni opseg razvijenog modela je bio prilično ograničen zbog njegovog linearnog karaktera, pa su u pomenutim radovima prikazani rezultati u opsegu u kojem su vrednosti pritiska u usisnom kolektoru bile između 50 i 60 kPa, a broj obrtaja između 1050 i 1130 min<sup>-1</sup>. U daljem istraživanju [22], Rizzoni i saradnici su koristili nelinearni model, zasnovan na paritetnim jednačinama, za generisanje pet reziduala. Posmatrane su neispravnosti senzora položaja leptira, senzora pritiska u usisnom kolektoru, senzora broja obrtaja motora i brizgača goriva na jednom stvarnom četvorocilindarskom motoru, radne zapremine 1.3 litra, koji je radio bez spoljašnjeg opterećenja. I u ovom slučaju su uspešno detektovane greške veće od 10%.

Gertler i saradnici su u publikacijama [23] i [24] prikazali rezultate simulacionog istraživanja, koje su izveli u okviru jednog projekta sa kompanijom General Motors. Koristili su pet reziduala u obliku linearnih paritetnih jednačina da bi detektovali neispravnosti senzora položaja leptira, EGR ventila, senzora pritiska u usisnom kolektoru, brizgača goriva, senzora broja obrtaja motora i lambda-sonde. Pomenuta kombinacija reziduala je bila u stanju da razlikuje sve neispravnosti, osim neispravnosti brizgača goriva i lambda-sonde. U naknadnim istraživanjima [25] Gertler i saradnici su primenili razvijeni princip i na GM-ovom V6 motoru na realnom vozilu. Isprobavanjem i off-line i on-line verzije svog dijagnostičkog sistema na realnom vozilu, ustanovili su da je jednostavnost primenjenih algoritama veoma značajna zbog ograničenih performansi on-board računara. Umesto pet linearnih, ovoga puta su koristili šest nelinearnih paritetnih jednačina za generisanje reziduala. Posmatrali su iste neispravnosti i utvrdili da njihov dijagnostički sistem uspešno detektuje greške na svim posmatranim uređajima, ali samo ako su one veće od 10%.

Nyberg i Nielsen [26] su razvili jedan sistem za dijagnostikovanje neispravnosti zasnovan na nelinearnom polu-fizičkom modelu. Pri tome su koristili kombinaciju nekoliko različitih metoda za generisanje reziduala. Razvijeni dijagnostički sistem je bio sposoban da detektuje i lokalizuje neispravnosti aktuatora leptira, senzora položaja leptira, masenog protokomera vazduha i senzora pritiska u usisnom kolektoru. Korišćenjem više metoda za generisanje reziduala dobili su fleksibilniji dijagnostički sistem sa boljim performansama, što su potvrdili eksperimentalnim ispitivanjem na stvarnom motoru.

Nyberg i Perkovic su u referenci [27] prikazali svoje istraživanje nezaptivenost usisnog sistema turboprehranjanog oto motora. Razvili su jedan modelski dijagnostički metod na bazi rekurzivnog algoritma najmanjih kvadrata, koji je bio u stanju da detektuje i veoma male neželjene otvore (prečnika do 2 mm) u usisnom sistemu, ispred i iza leptira. Osim toga razvili su i metod za rekonfigurisanje upravljačkog algoritma koji je mogao da neutrališe uticaj nezaptivenosti usisnog sistema na emisiju izduvnih gasova.

Conatser i saradnici [28] bavili su se dijagnostikovanjem neispravnosti sistema za upravljanje elektronskim leptirom. Oni su izvršili modelovanje tog sistema i primenili metodu paritetnih jednačina za generisanje reziduala, kako bi uporedili ulazne/izlazne vrednosti modela i realnog sistema. To im je omogućilo da izvrše detektovanje

neispravnosti u sistemu. Lokalizovanje neispravnosti u posmatranom sistemu izveli su korišćenjem metode procene parametara.

Franchek i ostali bavili su se detektovanjem, lokalizovanjem i procenom veličine neispravnosti koje se pojavljuju u usisnom sistemu motora SUS tokom rada na ustaljenom režimu [29]. Razmatrane su greška ofseta masenog protokomera vazduha i ulazak neželjenog vazduha u usisni sistem. Predložen je jedan dijagnostički pristup, zasnovan na statičkom modelu usisnog sistema, čije su ulazne veličine položaj leptira i broj obrtaja motora, a izlaz je maseni protok vazduha. Ovaj model se automatski adaptira, tako da vrednost njegove izlazne veličine odgovara izmerenoj vrednosti te veličine pri ustaljenom radu motora. Tokom pomenutog procesa adaptacije, menjaju se koeficijenti modela, a onda se, poređenjem koeficijenata originalnog i adaptiranog modela, generiše tzv. vektor greške koeficijenata. Intenzitet ovog vektora koristi se da se utvrdi da li postoji neka neispravnost, a njegov smer se koristi za određivanje vrste neispravnosti. Procena veličine neispravnosti realizovana je analiziranjem reziduala između stvarne izmerene vrednosti i modelovane vrednosti masenog protoka vazduha.

Sengupta i saradnici su se u [30] bavili detektovanjem ulaska neželjenog vazduha u usisni sistem i greške kalibrisanja masenog protokomera vazduha kod oto motora. Razvili su dva estimatora za detektovanje ovih neispravnosti i izvršili poređenje njihovih performansi. Prvi estimator je zasnovan na osrednjenom modelu motora i pomoću njega je modelovan maseni protok vazduha pored leptira, a drugi se bazira na ciklusnom modelu motora i izlazna veličina mu je maseni protok vazduha koji ulazi u cilindar. U oba estimatora su primenjene direktne fizičke relacije i prošireni Kalmanov filter za generisanje analitičke redundantnosti. Za testiranje performansi oba razvijena modela korišćeni su rezultati numeričke simulacije razmatranih neispravnosti pomoću komercijalnog softvera AMESim™, na jednom hipotetičkom jednocilindarskom motoru, koji radi na konstantnom broju obrtaja u otvorenoj sprezi sa lambda-sondom.

Ceccarelli i ostali razvili su dva različita nelinearna opservera za procenu veličine nezaptivenosti usisnog sistema dizel motora [31]. Predloženi modelski pristupi dijagnostikovanja neispravnosti zasnovani su na adaptivnim opserverima, od kojih jedan ima fiksno, a drugi varijabilno pojačanje. Oba predložena opservera su nelinearni i projektovani su na osnovu funkcija Ljapunova. Ovi adaptivni opserveri ne samo da omogućavaju procenu veličina koje se ne mogu direktno meriti, nego i nekih nepoznatih modelskih parametara, kao što je u ovom slučaju veličina koja je u direktnoj vezi sa neispravnošću koju treba detektovati, a to je prečnik neželjenog otvora u usisnom sistemu. Rezultati primene dva projektovana opservera na jedan četvorocilindarski dizel motor, pokazuju da se pomoću oba opservera mogu podjednako uspešno detektovati relativno mali otvori u usisnom sistemu, prečnika manjeg od 5 mm. Takođe, ovi opserveri procenjuju veličinu otvora sa greškom reda veličine 1mm, što je veoma dobra procena. Zbog jednostavnije strukture adaptivnog opservera sa fiksnim pojačanjem, autori daju prednost ovom opserveru u primeni u realnim aplikacijama.

Salehi i saradnici [32] su razvili jednu novu strategiju za otkrivanje isticanja vazduha iz usisnog sistema u delu između kompresora i leptira kod turboprehranjivanih oto motora. Strategija je primenljiva na motore bez masenog protokomera vazduha i zasniva se na estimatorima. U okviru te strategije, izvršeno je modelovanje dinamike rotora turbokompresora u sprezi sa dinamikom pritiska u interkuleru, a onda je, na osnovu dobijenog kompleksnog modela, projektovan estimator turbokompresora. Greška procene pritiska prehranjivanja od strane estimatora korišćena je kao rezidual u procesu detektovanja isticanja vazduha iz usisnog sistema. Predložena strategija je testirana u prisustvu dva tipa neispravnosti: isticanje vazduha iz usisnog sistema i greška u očitavanju pritiska prehranjivanja, jer postoji realna mogućnost da očitavanje niže vrednosti pritiska prehranjivanja bude prepoznato kao isticanje vazduha. Eksperimentalni rezultati su

pokazali da se pomoću predložene strategije uspešno mogu detektovati isticanja vazduha kroz otvore prečnika 5 mm i 7mm, dok greška očitavanja vrednosti pritiska prehranjivanja koja je za 3% niža od stvarne vrednosti, nije detektovana kao isticanje vazduha.

## 2.4.2 Radovi koji ilustruju razvoj i primenu modela zasnovanih na neuronskim mrežama

U radu [33] iz 2006. godine, Sangha i ostali predstavili su novi metod za on-board monitoring usisnog sistema oto motora. Suštinu ovog metoda čini adaptivni klasifikator neispravnosti baziran na neuronskim mrežama sa Gausovom radijalnom baznom funkcijom. Adaptivnost razvijenog klasifikatora ogleda se u tome što on uzima u obzir promene parametara motora, koje nastaju kao rezultat starenja ili promene eksploatacionih uslova. Ovaj klasifikator se može prilagoditi i razlikama između srodnih motora, koji pripadaju istoj familiji. Razmatrani su različiti nivoi četiri vrste neispravnosti: EGR ventil zaglavljnjen u nekom položaju, nezaptivenost usisnog sistema, pogrešno očitavanje pritiska u usisnom kolektoru i pogrešno očitavanje temperature u usisnom kolektoru. Adaptivni koncept funkcioniše tako što, kada klasifikator dijagnostikuje neku od pomenutih neispravnosti, dolazi do promene skupa podataka za obuku u smislu da se menjaju željene vrednosti na izlazu klasifikatora. Na osnovu ovih modifikovanih vrednosti željenih izlaza i trenutnih izmerenih vrednosti ulaznih veličina, izvršava se korekcija parametara neuronske mreže. Koeficijenti širine Gausove funkcije svakog skrivenog neurona koriguju se primenom metode opadajućeg gradijenta<sup>5</sup>, a težinski koeficijenti sinapsi između skrivenog i izlaznog sloja primenom rekurzivne metode najmanjih kvadrata<sup>6</sup>. Takođe, u ovom radu izvršeno je i poređenje performansi adaptivnog i neadaptivnog klasifikatora sa aspekta uspešnosti klasifikacije neispravnosti, kao i sa aspekta veličina primenjenih neuronskih mreža. Pokazalo se da je adaptivni klasifikator bolji sa oba aspekta. Inače, podaci za obuku i testiranje neuronske mreže oba klasifikatora dobijeni su pomoću osrednjenog modela motora, ali nije izvršena njihova eksperimentalna verifikacija.

Istraživanje iz [33] prošireno je upoređivanjem robusnosti adaptivnog i neadaptivnog klasifikatora, što je publikovano u [34]. Robusnost je utvrđivana praćenjem ponašanja klasifikatora u slučajevima, kada se pojave neke promene uzrokovane trenutnim uslovima eksploatacije motora ili njegovim starenjem. Konkretno, te promene su predstavljane različitim zakonitostima promene položaja leptira i spoljašnjeg opterećenja motora, kao i malim povećanjem radne zapremine motora (kao posledica habanja). Podaci za testiranje klasifikatora dobijeni su pomoću osrednjenog modela motora koji je modifikovan tako da obezbeđuje rad motora na nekom zadatom konstantnom broju obrtaja pri pojavi različitih neispravnosti u usisnom sistemu. Testiranje na tako dobijenim podacima pokazalo je superiornost adaptivnog u odnosu na neadaptivni klasifikator sa aspekta otpornosti na različite promenljive uslove rada motora.

Hamad i ostali razvili su u [35] jedan dijagnostički koncept za detektovanje neispravnosti kod automobilskih motora, koji se zasniva na jednoj nezavisnoj neuronskoj mreži. Podaci koji su korišćeni za obuku ove neuronske mreže nisu dobijeni merenjem na stvarnom motoru, već simulacijama realizovanim pomoću osrednjenog modela motora, čije ulazne veličine su ugao leptira i maseni protok goriva, a izlazne su broj obrtaja motora, pritisak i temperatura u usisnom kolektoru. Svih pet pobrojanih veličina su ulazne veličine neuronske mreže, ali se poslednje tri preuzimaju rekurzivno sa izlaza same neuronske mreže, što i čini ovu mrežu nezavisnom od osrednjenog modela motora. Na taj način je

---

<sup>5</sup> eng. Gradient Descent method

<sup>6</sup> eng RLS – Recursive Least Squares algorithm

postignuto to, da neuronska mreža uvek daje izlaze koji odgovaraju ispravnim podacima, pa su poređenjem tih izlaza sa izlazima osrednjenog modela motora generisani reziduali, koji su iskorišćeni za detektovanje neispravnosti. U ovom radu su posmatrane sledeće neispravnosti: nezaptivenost usisnog sistema (protok neželjenog vazduha 20% od ukupnog protoka kroz usisni kolektor), maseni protok goriva umanjen na 20% od normalnog, davač broja obrtaja očitava 10% veću vrednost, davač pritiska u usisnom kolektoru očitava 20% veću vrednost i davač temperature u usisnom kolektoru očitava 10% veću vrednost.

Nadovezujući se na istraživanje objavljeno u [35], Yu i ostali su u [36] predstavili jedan novi metod za lokalizovanje detektovanih neispravnosti davača broja obrtaja motora, davača pritiska i temperature u usisnom kolektoru, kao i nezaptivenosti usisnog kolektora i umanjenog doziranja goriva. Sposobnost izolovanja neispravnosti u ovom metodu zasniva se na još jednoj neuronskoj mreži, koja se pridodaje i deluje u sprezi sa ranije razvijenom nezavisnom neuronskom mrežom za detektovanje neispravnosti. Za razliku od neuronske mreže za detektovanje, koja je dinamička, neuronska mreža za izolovanje neispravnosti je statička. Kao ulazne veličine ona koristi rezidualne izlaznih veličina osrednjenog modela motora i izlaznih veličina neuronske mreže za detektovanje neispravnosti (broj obrtaja motora, pritisak u usisnom kolektoru i temperatura u usisnom kolektoru), a izlazne veličine su klasifikovane neispravnosti. Obe neuronske mreže su sa radijalnom baznom funkcijom.

P-C Chen [37] je razvio novi dijagnostički sistem za detektovanje ustrujavanja neželjenog vazduha u usisni sistem automobilskog motora. Ovaj dijagnostički sistem čine četiri podsistema od kojih jedan detektuje veličinu razmatrane neispravnosti u blizini sistema za ventilaciju kućišta motora, drugi detektuje veličinu razmatrane neispravnosti u zoni regulatora pritiska goriva, treći služi za detektovanje izvora razmatrane neispravnosti, a četvrti je sistem za odlučivanje. Podsystemi za detektovanje su bazirani na neuronskim mrežama bez povratnih sprega, koje su obučavane algoritmom sa prostiranjem greške unazad. Kao ulazne veličine neuronskih mreža korišćeni su: protok vazduha, dužina impulsa ubrizgavanja, pritisak u usisnom kolektoru i položaj leptira. Koristeći informacije, dobijene od tri podsistema za detektovanje, podsystem za odlučivanje vrši identifikaciju, kako mesta na kojem dolazi do ulaska neželjenog vazduha u usisni sistem, tako i veličine te neispravnosti. Različite veličine i različiti izvori neispravnosti simulirani su bušenjem dva, tri ili četiri otvora prečnika po 1mm u oblasti ventilacionog sistema kućišta motora ili u oblasti regulatora pritiska goriva. Eksperimenti, izvedeni na jednom Mitsubishi-jevom četvorocilindarskom motoru sa multipoint sistemom ubrizgavanja benzina, pokazali su da razvijeni dijagnostički sistem može uspešno da detektuje lokaciju i veličinu razmatrane neispravnosti, ali su merenja vršena samo u jednoj radnoj tački motora, na broju obrtaja od oko 1000 min<sup>-1</sup>.

Sangha i ostali u [38] bavili su se istraživanjem mogućnosti neuronskih mreža sa radijalnom baznom funkcijom (RBF<sup>7</sup>) u oblasti otkrivanja, prepoznavanja i korigovanja poznatih grešaka očitavanja različitih motorskih senzora, kao i detektovanja nepoznatih neispravnosti na motoru. Podaci za obuku i testiranje neuronskih mreža prikupljeni su pomoću jednog osrednjenog modela motora simuliranjem grešaka očitavanja: davača broja obrtaja motora, davača položaja leptira, davača pritiska u usisnom kolektoru i davača temperature u usisnom kolektoru. Sve pobrojane greške simulirane su kao 10% manje i 10% veće očitavanje u odnosu na tačnu vrednost merene veličine. Kao nepoznate neispravnosti, simulirane su sledeće: nezaptivenost usisnog sistema (protok neželjenog vazduha 20% od ukupnog protoka kroz usisni kolektor), EGR ventil zaglavljnjen u zatvorenom položaju, prebogata smeša (koeficijent viška vazduha 0.73) i presiromašna smeša (koeficijent viška vazduha 1.68). Za detektovanje i prepoznavanje svih pomenutih

---

<sup>7</sup> eng. Radial Basis Function



neispravnosti generisana je jedna RBF neuronska mreža, a još tri su generisane u svrhu akomodacije grešaka davača broja obrtaja, davača pritiska u usisnom kolektoru i davača temperature u usisnom kolektoru. Akomodacija grešaka svakog od tri davača vršena je pomoću odgovarajuće neuronske mreže, predviđanjem vrednosti koju bi trebalo da da neispravan davač, na osnovu trenutnih i prethodne tri vrednosti koje daju preostala dva ispravna davača.

### 2.4.3 Ostali radovi od značaja za predmetno istraživanje

Nyberg i saradnici su pred kraj prošlog i početkom ovog veka objavili određen broj publikacija, u kojima su u značajnoj meri primenjivali statističke metode u dijagnostikovanju neispravnosti. Godine 1999. Nyberg je projektovao kompletan dijagnostički sistem za usisni sistem oto motora i u publikacijama [1] i [39] je predstavio proceduru njegovog projektovanja, uključujući odabir i podešavanje postupka testiranja hipoteza, kao i veličina koje će biti testirane. Posmatran je veći broj neispravnosti, kao što su postojanje neželjenog otvora u usisnom sistemu (ispred i iza leptira) i neispravnosti senzora pritiska u usisnom sistemu (ispred i iza leptira), senzora položaja leptira i masenog protokomera vazduha. Uspešno su dijagnostikovane sve neispravnosti na jednom stvarnom turboprehranjivanom oto motoru.

Dalje, 2000. godine, Nyberg [40] je istraživao dve različite metode za dijagnostikovanje neželjenih otvora u usisnom sistemu automobilskog motora. Prva se zasniva na poređenju izmerenog i procenjenog protoka vazduha, a druga na proceni površine poprečnog preseka neželjenog otvora. Ove dve metode su poređene korišćenjem statističkih metoda testiranja hipoteza. Pokazalo se, i teorijski i na stvarnom turboprehranjivanom oto motoru, da metoda, bazirana na proceni veličine otvora u usisnom sistemu, daje bolje rezultate, pa je predložena kao bolji izbor u slučaju da se razmatra samo nezaptivenost usisnog sistema. Međutim, ako treba dijagnostikovati još neke neispravnosti, onda je uspešnija metoda bazirana na poređenju procenjenog i izmerenog protoka vazduha.

Prema Nyberg-u, modeli neispravnosti su moćan alat za dijagnostikovanje svih vrsta neispravnosti. On je u publikaciji [41] pokazao da metoda testiranja strukturisanih hipoteza omogućava uspešno dijagnostikovanje neispravnosti koje se modeluju kao odstupanje konstantnih parametara, odstupanje proizvoljnih signala, ili kao njihova kombinacija. Ukazao je i da modelovanje neispravnosti povećava mogućnost lokalizovanja različitih neispravnosti, jer ako dve neispravnosti deluju na različite načine na istu komponentu sistema, one su pojedinačno prepoznatljive.

Godine 2004. Nyberg i Stutte su u publikaciji [42] za jedan motor razvili dijagnostički sistem, značajno komplikovaniji od svih koji su razmatrani u prethodnim radovima iz ove oblasti. U pitanju je bio Mercedesov automobilski dizel motor, sa EGR sistemom i turbo-kompresorom varijabilne geometrije. Autori su projektovali matematički model motora relativno visoke tačnosti (prosečna greška modela manja od 3%), što su potvrdili na stvarnom vozilu. Korišćenjem dobijenog modela, i ranije razvijenog dijagnostičkog okvira zasnovanog na testiranju strukturisanih hipoteza, razvili su dijagnostički sistem, koji je pokazao veoma dobre performanse. Razmatrane su greške masenog protokomera vazduha i senzora pritiska u usisnom kolektoru, kao i nezaptivenost usisnog sistema i neispravnost EGR ventila (zaglavljene u zatvorenom položaju). Greške senzora su simulirane u Matlab-u, korišćenjem stvarnih ispravnih podataka, a neispravnosti komponenti su realizovane na stvarnom motoru. Zanimljivo je da su tokom ovog istraživanja Nyberg i Stutte primenili i metodu fizičke redundantnosti, jer su na svom eksperimentalnom postrojenju instalirali dodatne senzore pritiska i temperature u usisnom kolektoru.

U okviru publikacije [43], Capriglione i saradnici izložili su jednu proceduru za on-board dijagnostikovanje i neutralisanje grešaka senzora, koji su obično montirani u vozilima

javnog prevoza. Predložena procedura je zasnovana na analitičkoj redundantnosti i obuhvata akviziciju podataka sa senzora, generisanje reziduala, zatim detektovanje grešaka, prepoznavanje neispravnog senzora i na kraju neutralisanje njegove neispravnosti. Detektovanje neispravnosti i identifikovanje neispravnog senzora je bazirano na „if-then“ pravilima. Neutralisanje neispravnosti senzora izvodi se zamenom izlaznog signala tog senzora procenjenom vrednošću merene veličine, koja se određuje analitički.

Tema rada [44] Isermann-a i Schwarte-a bila je razvoj jednog metoda za dijagnostikovanje neispravnosti u usisnom sistemu dizel motora. Osnova ove metode bila je kombinacija jednog polu-fizičkog modela procesa u usisnom sistemu i dinamičke neuronske mreže. Na bazi ovog kombinovanog modela izvedene su paritetne jednačine, pomoću kojih su generisani reziduali. Ovi reziduali su praktično predstavljali simptome razmatranih neispravnosti i bili su ključni za njihovo dijagnostikovanje. Predloženi koncept je proveren u realnom vremenu na jednom dvolitarskom turbo-prehranjivanom dizel motoru u laboratorijskim uslovima. Eksperimentalna provera je pokazala da su zahtevi za računarskim resursima pri primeni ovog koncepta umereni i ne predstavljaju smetnju za on-board primenu u realnom vremenu.

Rad [45] Kimmich-a i saradnika iz 2005. godine pokazuje razvoj jednog sistematičnog modela za dijagnostikovanje neispravnosti u usisnom sistemu i gorivnom sistemu dizel motora. Korišćeno je modelovanje relevantnih fizičkih veličina pomoću pet lokalnih linearnih neuronskih mreža. Na osnovu toga, napisano je pet paritetnih jednačina, pomoću kojih su generisani reziduali. Obrada signala i algoritmi za modelsko detektovanje neispravnosti implementirani su u Matlab/Simulink okruženju, a eksperimentalna istraživanja su izvršena na Opel-ovom četvorocilindarskom motoru na probnom stolu i u samom vozilu, primenom razvijenog dijagnostičkog modela u realnom vremenu.

Chen, u radu [46], predstavlja jedan neuro-fazi dijagnostički sistem za detektovanje greške protokomera vazduha pri radu benzinskog motora na praznom hodu. Ovaj dijagnostički sistem bazira se na Takagi-Sugeno fazi-modelu i formulisan je na ulazno-izlaznim relacijama između simptoma i neispravnosti. Parametri sistema se podešavaju kroz proces obuke na osnovu eksperimentalno dobijenih podataka sa davača broja obrtaja motora, protokomera vazduha, senzora položaja leptira, i podataka o dužini impulsa ubrizgavanja i ugla pretpaljenja. U obuci neuronske mreže primenjeni su metod najbržeg opadanja gradijenta i algoritam prostiranja greške unazad. Predloženi dijagnostički sistem sastoji se iz dva dela: prvi treba da oceni da li postoji greška protokomera vazduha, a drugi treba da proceni veličinu greške. Verifikacija predloženog dijagnostičkog pristupa izvršena je u laboratorijskim uslovima na četvorocilindarskom Mitsubishi-jevom benzinskom motoru, simuliranjem pogrešnog signala koji ECU dobija sa protokomera

U Antory-jevom radu [47] iz 2007. godine predstavljen je jedan dijagnostički model baziran na podacima dobijenim iz mernih signala sa senzora i aktuatora koji su prisutni u modernim automobilima, ali i signala sa mernih uređaja na probnom stolu korišćenom za laboratorijska ispitivanja motora. Značaj predloženog modela je u tome što on detaljno analizira sve eksperimentalne podatke, a onda metodom analize glavnih komponenti<sup>8</sup>, preuzima samo važne informacije iz pojedinih originalnih mernih signala i koristi ih u dijagnostičke svrhe. U radu je izvršeno i poređenje dijagnostičkih performansi ovog modela sa konvencionalnim analitičkim modelom, na primerima otvora prečnika 2, 4 i 6 mm u usisnom kolektoru jednog Volkswagen-ovog turbo-prehranjivanog dizel motora. Rezultati su bili na strani novopredložene metode, što je bilo posebno vidljivo u slučaju najmanjeg otvora.

---

<sup>8</sup> eng. Principal Component Analysis

Wang i saradnici su se u [48] bavili istraživanjem detektovanja neželjenog otvora u usisnom kolektoru automobilskog motora tokom njegovog rada na prelaznim režimima. Oni su predložili jedan novi pristup u detektovanju pomenute neispravnosti u usisnom sistemu motora, koji predstavlja kombinaciju analitičkog modela, neuronskih mreža i lokalnog pristupa. Analitički model za izračunavanje masenog protoka vazduha iskorišćen je da bi se umanjila kompleksnost neuronskog modela, za koji je korišćena autoasocijativna neuronska mreža. Zatim je na neuronsku mrežu primenjen lokalni pristup, kako bi se generisala tzv. Hotelling-ova  $T^2$  statistika, neophodna za detektovanje neispravnosti. Primena analitičkog modela u predloženom pristupu olakšala je i generisanje  $T^2$  statistike u smislu smanjenja kompjuterskog opterećenja prilikom njenog računanja. Validacija novog pristupa izvršena je sa uspehom na podacima dobijenim sa jednog Nissan-ovog benzinskog motora.

Qadeer i ostali [49] su rešavali problem detektovanja ulaska neželjenog vazduha u usisni kolektor u ranoj fazi, kako bi se predupredile ozbiljnije posledice, koje mogu da nastanu zbog ove neispravnosti. Oni su ponudili jedan koncept za predviđanje i detektovanje ove neispravnosti, koji je zasnovan na diskretnom skrivenom Markovljevom modelu (HMM – Hidden Markov Model) i to je prvi put da je ovaj model predložen u oblasti dijagnostike usisnog sistema automobilskih atmosferskih motora. HMM je jedan statistički alat koji se koristi za klasifikovanje podataka, obradu signala i dijagnostikovanje grešaka u inženjerskim sistemima. Osnovna prednost predloženog koncepta je ta, što koristi samo podatke od davača broja obrtaja i davača pritiska u usisnom kolektoru, koje može da dobije preko OBD II skenera, pa se relativno lako može integrisati u bilo koji softver koji se koristi na tom uređaju. U predloženom konceptu, posmatrana su četiri skrivena stanja, u zavisnosti od prečnika otvora kroz koji ulazi neželjeni vazduh: 0mm, 2.5mm, 5mm i 10mm, a merenja su vršena pri radu motora na praznom hodu.

# 3 PRIMENA VEŠTAČKIH NEURONSKIH MREŽA U PREPOZNAVANJU UZORAKA

## 3.1 Uvodna razmatranja

Na početku razmatranja o neuronskim mrežama trebalo bi razjasniti neke terminološke stvari. Kada se u tehničkim disciplinama govori o neuronskim mrežama, uglavnom se misli na veštačke neuronske mreže. Veštačke neuronske mreže su sistemi za obradu informacija koji imaju neke zajedničke karakteristike sa ljudskim mozgom. Originalne neuronske mreže, sačinjene od specijalnih nervnih ćelija (bioloških neurona), koje se nalaze u mozgu čoveka, ovde će se zvati biološkim neuronskim mrežama. Prvopomenute, veštačke neuronske mreže, do kraja ovog rada imaće naziv samo „neuronske mreže“, jer je takav, skraćeni naziv odomaćen u ovoj oblasti veštačke inteligencije. Kada je u pitanju priroda neuronske mreže, još jedna dilema se često javlja kod ljudi koji se ne susreću svakodnevno sa ovim pojmom: da li je neuronska mreža neki specijalni komad računarskog hardvera ili je to računarski program? U osnovi, neuronske mreže su matematički modeli za obradu informacija, koji principijelno ne moraju da imaju veze ni sa računarskim softverom, ni sa hardverom. Međutim, otkako je krajem prošlog veka počela ekspanzija informacionih tehnologija, praktična primena neuronskih mreža je direktno vezana za razvoj ove oblasti. Neuronske mreže se danas uglavnom implementiraju u vidu softverskih modela, a bilo je i nekih pokušaja da se neuronske mreže realizuju i direktno hardverski, u vidu specijalnih čipova [50], [51].

Univerzalno prihvaćena definicija neuronske mreže ne postoji, ali jedna od najčešće navođenih u literaturi je Haykin-ova [52] definicija: „Neuronska mreža je paralelni distribuirani procesor, koji ima prirodnu sposobnost čuvanja i korišćenja iskustvenog znanja, i čija se sličnost sa mozgom ogleda kroz dva svojstva:

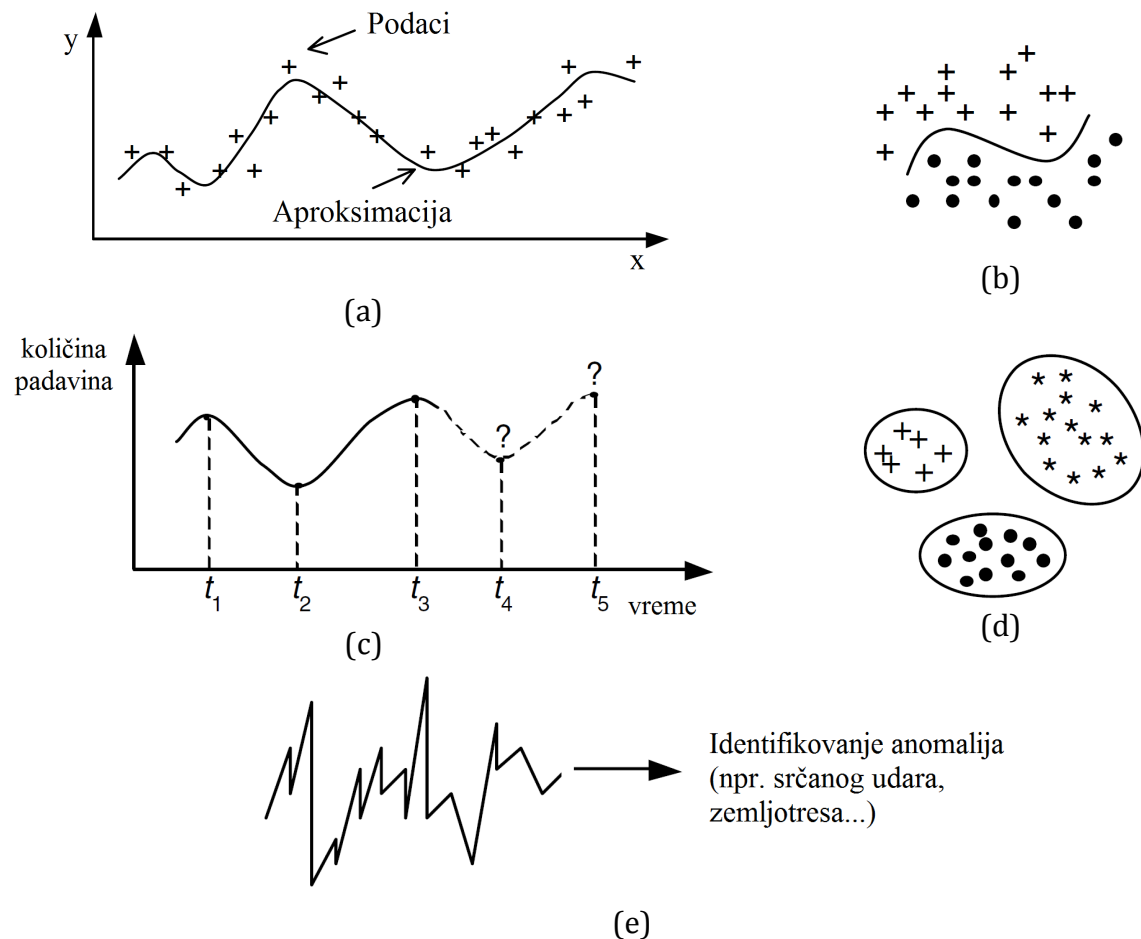
- mreža stiče znanje kroz proces učenja i
- znanje se čuva u vezama neurona (sinaptičkim težinama)“.

Proces učenja i uloga sinaptičkih težina u tom procesu biće razjašnjeni kasnije, kada će se o njima posebno govoriti, a za sada treba znati da neuronsku mrežu čini skup elementarnih jedinica za obradu podataka koje se nazivaju neuronima, a međusobno su, na određeni način, povezane takozvanim sinapsama.

Neuronske mreže stiču znanje na osnovu primera. Baš kao što neko dete na osnovu konkretnih primera pasa i mačaka nauči da razlikuje ove dve vrste životinja, čak i kada vidi recimo nekog psa ili mačku koje do tada nikada nije videlo. Slično tome, i neuronska mreža, posle obuke na konkretnim podacima iz okoline, treba da „reaguje“ na zadovoljavajući način na nove podatke koji nisu korišćeni u procesu obuke, tj. treba da ima sposobnost dobre generalizacije. O generalizaciji će kasnije biti malo više reči. Koja se to reakcija neuronske mreže smatra zadovoljavajućom, zavisi od vrste problema na koji se neuronska mreža primenjuje. Pošto je tema ovog rada dijagnostika, može se reći da bi zadovoljavajuća reakcija u tom slučaju bila, da neuronska mreža detektuje da se desila neka neispravnost, kao i da prepozna koja je to neispravnost. Drugim rečima, neuronska mreža bi trebalo da klasifikuje neispravnost u neku od unapred definisanih kategorija.

Dakle, najvažnije svojstvo neuronskih mreža je njihova sposobnost da kroz primere nauče kako se neki sistem ponaša, a ne da matematički detaljno opisuju karakteristike tog sistema. Ovo njihovo svojstvo je veoma korisno kada ne postoje detaljne i potpune

informacije o sistemu, nego samo skupovi njegovih ulaznih i izlaznih podataka. A takvih sistema, koji nisu i ne mogu biti potpuno matematički opisani, je veliki broj u našem okruženju. Zato se neuronske mreže primenjuju u gotovo svakoj oblasti ljudske delatnosti, a ovde će biti navedene samo neke vrste zadataka koji se najuspešnije rešavaju pomoću neuronskih mreža. Takvi su, na primer: aproksimacija složenih nelinearnih funkcija, određivanje pripadnosti uzoraka nekoj od unapred definisanih klasa (prepoznavanje uzoraka), analiza signala u vremenu, svrstavanje podataka ili objekata u nepoznate kategorije (kategorizacija), predviđanje trenda neke promenljive u budućnosti. Radi ilustracije, na Sl. 3.1 data je uprošćena vizuelna prezentacija pobrojanih zadataka.



Sl. 3.1 Ilustracija najčešćih zadataka neuronskih mreža: (a) Aproksimacija funkcija; (b) Prepoznavanje uzoraka; (c) Prognoziranje trenda; (d) Kategorizacija uzoraka; (e) Analiza signala u vremenu. [53]

S obzirom na raznovrsne zadatke koji se mogu rešavati pomoću neuronskih mreža, tokom više od 60 godina njihovog postojanja razvijen je veliki broj vrsta neuronskih mreža, različitih po raznim aspektima. Pošto neuronske mreže nisu centralna tema istraživanja, diskusija o njihovoj raznovrsnosti će u ovom radu izostati. Imajući u vidu da je u ovom istraživanju jedan od ciljeva prepoznavanje neispravnosti i njihovo klasifikovanje u neku od unapred definisanih grupa, ovde će interesovanje biti zadržano na onim neuronskim mrežama koje su se pokazale kao najbolje za rešavanje ovakve vrste zadataka, a to su višeslojne neuronske mreže bez povratnih sprega<sup>9</sup>. Konkretno, tokom izvođenja ovog istraživanja isprobane su dve vrste ovih mreža: višeslojni perceptron<sup>10</sup> i neuronska mreža

<sup>9</sup> eng. *multilayer feed-forward neural networks*

<sup>10</sup> eng. *multilayer perceptron*

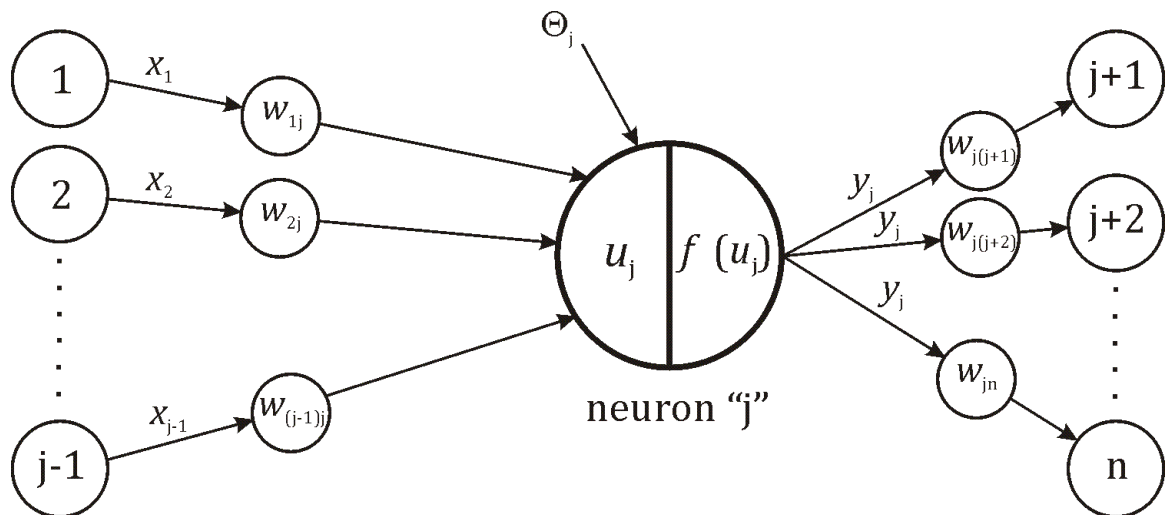
sa radijalnom baznom funkcijom <sup>11</sup>. Višeslojni perceptron se, inače, smatra standardnom neuronskom mrežom bez povratnih sprega, tako da, kada se kaže „neuronska mreža bez povratnih sprega“, misli se na višeslojni perceptron. U daljem tekstu će ova dva pojma biti korišćena ravnopravno. Rezultati, dobijeni u preliminarnoj fazi ovog istraživanja, izuzetno su dobri kada je u pitanju višeslojni perceptron, ali se to ne može reći za neuronske mreže sa radijalnom baznom funkcijom. Otuda će fokus u daljem tekstu biti usmeren na standardnu neuronsku mrežu bez povratnih sprega, tj. na višeslojni perceptron, a neuronske mreže sa radijalnom baznom funkcijom će biti izuzete iz daljeg razmatranja.

Funkcionisanje neuronske mreže određeno je načinom na koji su neuroni uzajamno povezani u okviru te mreže, jačinama veza između neurona, kao i obradom informacija koja se izvršava u okviru svakog neurona. O svakom od ovih bitnih segmenata biće više reči kasnije, jer, da bi se te stvari razumele, najpre se treba upoznati sa matematičkim modelom osnovnog gradivnog elementa svake neuronske mreže – neuronom.

## 3.2 Matematički model neurona

Šematski prikaz osnovnih elemenata jednog neurona, u kojem se vrši obrada podataka, dat je na Sl. 3.2. U centralnom delu slike je neki proizvoljni neuron  $j$ , na čijem primeru će biti opisana suštinska svojstva neurona.

Svaki neuron ima svoje unutrašnje stanje, nazvano aktivacija ili nivo aktivacije, koje se u vidu izlaznog signala prenosi prema drugim neuronima, povezanim sa tim neuronom, ili prema izlazu iz neuronske mreže [54]. Konkretno, aktivacija neurona  $j$  na Sl. 3.2 označena je sa  $y_j$ , a aktivacije neurona  $1, 2, \dots, j-1$ , označene su sa  $x_1, x_2, \dots, x_{j-1}$ , respektivno. Važno je istaći da jedan neuron u jednom vremenskom trenutku može da ima samo jedan izlazni signal, iako taj signal može emitovati prema većem broju drugih neurona. Tako, na primer, neuron  $j$  na Sl. 3.2 emituje svoj izlazni signal  $y_j$  prema svakom od neurona  $j+1, j+2, \dots, n$ .



Sl. 3.2 Šematski prikaz modela neurona

Veza ili sinapsa između svaka dva neurona ima jednu karakteristiku, koja je od suštinske važnosti za kompletnu neuronsku mrežu, kako je rečeno u odeljku 3.1, u definiciji Haykin-a. Ta karakteristika je jačina veze, koja se kvantifikuje realnim brojevima, poznatim pod nazivom *težinski koeficijenti* ili *sinaptičke težine*. Na Sl. 3.2, težinski koeficijenti veza između neurona  $1, 2, \dots, (j-1)$  sa jedne strane i neurona  $j$  sa druge strane, označene su sa  $w_{1j}, w_{2j}, \dots, w_{(j-1)j}$ , respektivno. Isto tako, težinski koeficijenti veza između neurona  $j$  i neurona

<sup>11</sup> eng. *Radial Basis Function Neural Network*

$(j + 1), (j + 2), \dots, n$  označene su sa  $w_{j(j+1)}, w_{j(j+2)}, \dots, w_{jn}$ , respektivno. Ukoliko je vrednost težinskog koeficijenta sinapse između dva neurona jednaka nuli, to znači da ta dva neurona nisu povezana.

Ukupni neto ulaz u neuron  $j$ , označen na slici sa  $u_j$ , najčešće se dobija kao suma signala  $x_i$  ( $i=1, \dots, j-1$ ) koje drugi neuroni emituju prema neuronu  $j$ , ponderisanih težinskim koeficijentima  $w_{ij}$ , umanjena za veličinu praga aktivacije neurona  $\Theta_j$ :

$$u_j = \sum_{i=1}^{j-1} x_i \cdot w_{ij} - \Theta_j. \quad (3.1)$$

Parametar  $\Theta_j$  se često ni ne prikazuje samostalno pri modelovanju neurona, nego se zamenjuje dodatnim ulaznim signalom ( $x_0 = 1$ ), koji se ponderiše težinskim koeficijentom  $w_{0j} = -\Theta_j$ , tako da jednačina (3.1) dobija kompaktniji oblik

$$u_j = \sum_{i=0}^{j-1} (x_i \cdot w_{ij}). \quad (3.2)$$

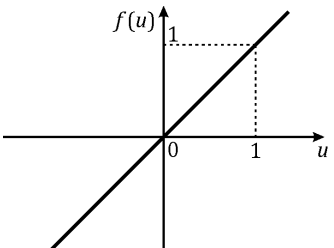
Funkcija  $u_j$  nije obavezno linearna, ali u najvećem broju slučajeva jeste takva, pa ostale varijante ovde neće biti pominjane.

Određivanje neto ulaza je prva faza obrade podataka u neuronu  $j$ . Druga faza je određivanje aktivacije  $y_j$  ovog neurona, što se ostvaruje primenom takozvane aktivacione funkcije  $f$  na njegov ukupni neto ulaz  $u_j$ , prema izrazu

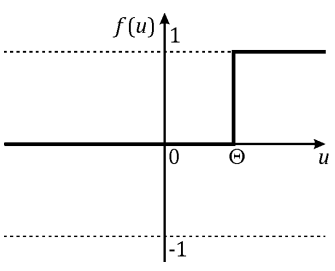
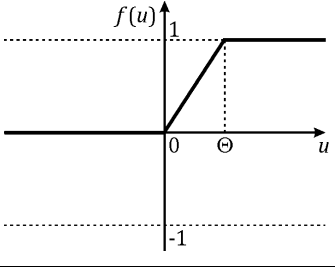
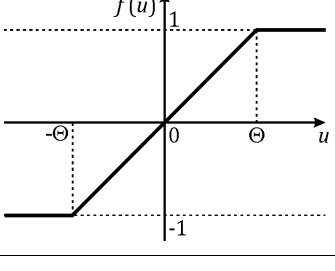
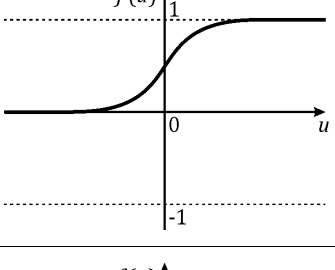
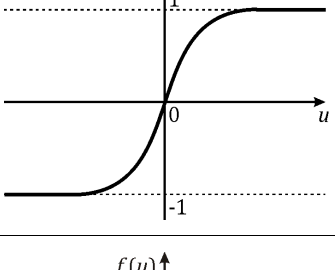
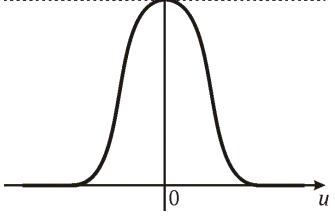
$$y_i = f(u_j). \quad (3.3)$$

U zavisnosti od problema koji se rešava neuronskom mrežom, kao i mesta koje konkretni neuron zauzima u neuronskoj mreži, aktivaciona funkcija neurona može imati različite oblike. Neke od funkcija koje se najčešće primenjuju u ove svrhe prikazane su u *Tab. 3.1*. Identička funkcija ne menja vrednost argumenta funkcije, pa se koristi uglavnom samo za preuzimanje podataka iz okruženja, jer u tom slučaju nije potrebna nikakva dodatna obrada podataka u neuronu. Odskočna funkcija je specifična po tome što se koristi za modelovanje neurona sa diskretnim vrednostima aktivacije. Ostale funkcije iz *Tab. 3.1* su karakteristične po tome što su njihovi kodomeni ograničeni u relativno uskim intervalima (između 0 i 1 ili između -1 i 1), što je svojstveno i signalima bioloških neurona [53]. U većini slučajeva, pri modelovanju realnih složenih sistema, koji se često karakterišu visokim nivoom nelinearnosti, linearne aktivacione funkcije ne mogu da daju zadovoljavajuće rezultate. Nasuprot njima, sigmoidne funkcije i Gausova funkcija, kao nelinearne, pokazale su se kao veoma korisne pri rešavanju velikog broja različitih realnih problema.

*Tab. 3.1 Najčešći oblici aktivacione funkcije*

Naziv funkcije	Oblik grafika	Matematički oblik
Identička		$f(u) = u$

Tab. 3.1 (Nastavak)

Odskočna		$f(u) = \begin{cases} 0, & u < 0 \\ 1, & u \geq 0 \end{cases}$
Linearna ograničena binarna		$f(u) = \begin{cases} 0, & u < 0 \\ \frac{1}{\theta} \cdot u, & 0 \leq u < \theta \\ 1, & u \geq \theta \end{cases}$
Linearna ograničena bipolarna		$f(u) = \begin{cases} 0, & u < -\theta \\ \frac{1}{\theta} \cdot u, & -\theta \leq u < \theta \\ 1, & u \geq \theta \end{cases}$
Sigmoidna binarna <sup>12</sup>		$f(u) = \frac{1}{1 + e^{-\sigma u}}$
Sigmoidna bipolarna <sup>12</sup>		$f(u) = \frac{1 - e^{-\sigma u}}{1 + e^{-\sigma u}}$
Gausova <sup>13</sup>		$f(u) = e^{-\left(\frac{u}{\rho}\right)^2}$

<sup>12</sup> Parametrom  $\sigma$  u jednačini određuje se nagib strmog dela grafika funkcije<sup>13</sup> Parametrom  $\rho$  u jednačini određuje se širina oblasti ispod grafika funkcije

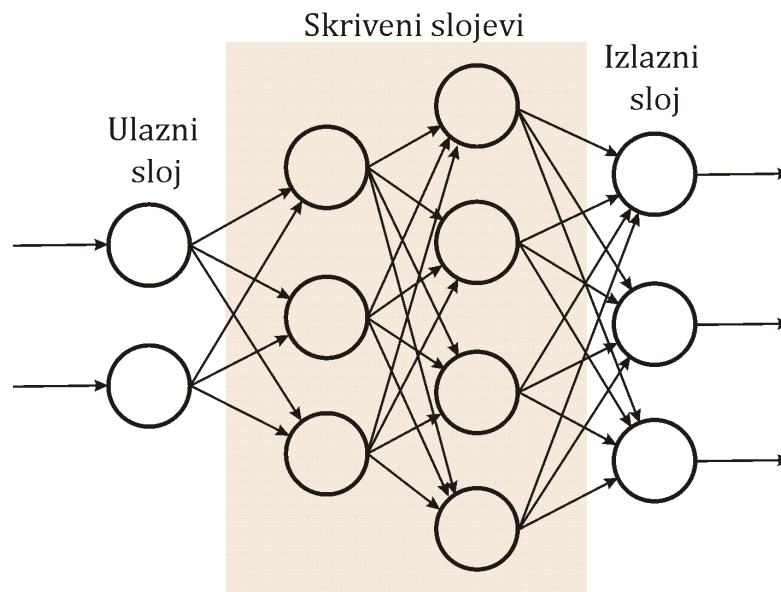


### 3.3 Struktura neuronske mreže bez povratnih sprega

Struktura neke neuronske mreže određena je njenom topologijom i aktivacionim funkcijama svih njenih neurona.

Kada se kaže „topologija“ neuronske mreže, misli se na ukupan broj neurona i način na koji su oni u toj mreži međusobno povezani. Osim ukupnog broja neurona, za topologiju neuronske mreže bitan je i broj neurona preko kojih se ostvaruje interakcija neuronske mreže sa spoljašnjim okruženjem. Neuronska mreža interaguje sa spoljašnjim okruženjem tako što iz okruženja dobija ulazne signale, a saopštava mu izlazne signale. Neuroni, koji primaju signale od spoljašnjeg okruženja zovu se ulazni, a oni koji šalju signale u okruženje su izlazni neuroni. U okviru neuronske mreže najčešće postoje i neuroni, koji nisu u direktnoj vezi sa okolinom i oni se nazivaju skrivenim neuronima. U zavisnosti od toga da li su i kako povezani sa okolinom, neuroni se grupišu u odgovarajuće slojeve sa logičnim nazivima: ulazni, skriveni i izlazni. Pri tome, treba istaći da neuronska mreža ne mora da ima ni jedan skriveni sloj, a može da ih ima i nekoliko. Teorijski, kod neuronskih mreža bez povratnih sprega moguća je veza između bilo koja dva neurona, pod uslovom da ne pripadaju istom sloju. Međutim, u praksi se ipak ređe susreće tolika sloboda u međusobnom povezivanju neurona kod ove vrste neuronskih mreža. Uglavnom su neuroni povezani sa neuronima iz susjednih slojeva. Generalno, svi neuroni koji pripadaju jednom sloju, ponašaju se na isti način – imaju istu šemu povezivanja sa ostalim neuronima i istu aktivacionu funkciju [54]. Drugim rečima, ako je bilo koji neuron iz jednog sloja povezan sa nekim neuronom iz drugog sloja, onda je svaki neuron iz prvopomenutog sloja povezan sa svakim neuronom iz drugog sloja. Ukoliko ovo važi za sve slojeve neuronske mreže, onda se ona naziva *potpuno povezanom*, a ako izostaje veza između bilo koja dva neurona u susjednim slojevima, mreža je *delimično povezana* [55].

Na Sl. 3.3 je šematski prikazana topologija jedne potpuno povezane višeslojne neuronske mreže bez povratnih sprega, koja ima dva skrivena sloja neurona. Konfiguracija neuronske mreže sa Sl. 3.3 skraćeno se označava sa 2-3-4-3, što znači da u ulaznom sloju ima 2 neurona, u prvom skrivenom sloju 3, u drugom skrivenom sloju 4 i u izlaznom sloju 3 neurona.



Sl. 3.3 Šematski prikaz topologije jedne potpuno povezane višeslojne neuronske mreže bez povratnih sprega

Inače, brojevi neurona u ulaznom i izlaznom sloju određeni su načinom na koji je definisan problem koji se želi rešiti pomoću neuronske mreže. Tačnije, broj ulaznih neurona jednak

je broju spoljašnjih ulaznih veličina, a broj izlaznih neurona poklapa se sa brojem izlaznih veličina neuronske mreže. Tako, na primer, ako se pomoću neuronske mreže želi aproksimirati funkcija  $z = f(x, y)$ , onda će ta mreža imati dva ulazna neurona (po jedan za vrednosti nezavisno promenljivih  $x$  i  $y$ ) i jedan izlazni neuron (za vrednost zavisno promenljive  $z$ ). Može se reći da je priroda problema aproksimacije funkcija takva, da se za njegovo rešavanje koriste uglavnom neuronske mreže sa jednim izlaznim neuronom. Međutim, pri rešavanju nekih drugih problema, na primer problema razvrstavanja uzoraka u  $N$  različitih klasa, obično se formiraju neuronske mreže sa  $N$  izlaznih neurona, tako da svakoj klasi odgovara po jedan izlazni neuron. U tom slučaju, aktivacija nekog izlaznog neurona može da ima binarne vrednosti. Vrednost 1 označava da konkretni uzorak, koji se trenutno obrađuje u neuronskoj mreži, pripada klasi kojoj odgovara taj izlazni neuron, a vrednost 0, da konkretni uzorak ne pripada toj klasi.

Dakle, što se tiče broja ulaznih i izlaznih neurona pri formiranju neuronske mreže, situacija je potpuno jasna. Zato se pitanje određivanja topologije neuronske mreže pri njenom dizajniranju svodi na određivanje broja skrivenih neurona i broja slojeva u koje će skriveni neuroni biti raspoređeni. Broj skrivenih neurona je veoma važan parametar u strukturi neuronske mreže, jer ako nije dobro odabran, performanse neuronske mreže nisu zadovoljavajuće. Ukoliko je broj skrivenih neurona nedovoljan, mreža ne može da nauči dovoljno detalja o modelovanom sistemu, pa će greška modelovanja biti velika. Sa druge strane, ako je broj skrivenih neurona preveliki, mreža će naučiti i neke beznačajne detalje, što se negativno odražava na njenu sposobnost da generalizuje.

U suštini, topologija neuronske mreže koja je najbolja za rešavanje nekog određenog problema obično se ne može odrediti, iako su u tom pogledu do sada uloženi određeni istraživački naponi, kao što se može videti u referencama [53], [56], [57], [58] i [59]. Do sada je razvijen relativno veliki broj algoritama za određivanje optimalnog broja skrivenih neurona i oni se uglavnom baziraju na dinamičkom kreiranju topologije neuronske mreže, tj. na dodavanju ili odstranjivanju skrivenih neurona tokom procesa obuke. Međutim, svi ovi algoritmi mogu se primeniti u nekim situacijama i uz određena ograničenja, jer ni jedan od njih ne uzima u obzir sve faktore od kojih zavisi optimalan broj skrivenih neurona u mreži. Najvažniji faktori koji utiču na određivanje optimalnog broja skrivenih neurona su :

- broj ulaznih i izlaznih neurona,
- broj uzoraka za obuku,
- složenost zadatka koji neuronska mreža treba da realizuje,
- aktivaciona funkcija skrivenog neurona i
- algoritam za obuku neuronske mreže.

Zbog ovako velikog broja uticaja, najpouzdaniji način da se odredi broj skrivenih neurona koji je najbolji za rešavanje konkretnog problema jeste pomoću metode „probe i greške“<sup>14</sup>. Drugim rečima, za neuronske mreže sa različitim brojem skrivenih neurona procenjuje se greška generalizacije i odabira se neuronska mreža sa minimalnom greškom. Pri tome se kreće od manjeg broja skrivenih neurona, a onda se taj broj povećava dok se ne dobiju zadovoljavajući rezultati ili dok se ne dostigne nivo, kada dalje povećanje broja skrivenih neurona ne dovodi do značajnih poboljšanja performansi neuronske mreže.

Što se tiče broja skrivenih slojeva, mogućnost izbora je mnogo manja, nego u slučaju broja neurona. Hornik i ostali [60], a zatim i Stinchcombe i ostali [61] ustanovili su da je neuronska mreža bez povratne sprege, koja ima samo jedan skriveni sloj neurona, sposobna da aproksimira bilo koju funkciju iz jednog konačnog višedimenzionalnog prostora u drugi, sa bilo kojom željenom tačnošću, pod uslovom da je aktivaciona funkcija tog skrivenog sloja nelinearna i da je u skrivenom sloju dovoljan broj neurona. Ova tvrdnja se sa problema aproksimacije može proširiti i na problem klasifikacije uzoraka [62].

---

<sup>14</sup> eng. *trial and error*

Međutim, za sada ne postoji teorija koja će tačno utvrditi broj neurona u skrivenom sloju, koji je neophodan da bi se aproksimirala bilo koja funkcija ili da bi se izvršila odgovarajuća klasifikacija.

Iako neuronska mreža sa jednim skrivenim slojem predstavlja univerzalni aproksimator, neka istraživanja su pokazala da se performanse neuronske mreže mogu poboljšati dodavanjem još jednog skrivenog sloja neurona [63], [64]. Neuronske mreže sa više od dva skrivena sloja se veoma retko primenjuju. Mogu se izuzetno susresti u nekim specijalnim aplikacijama [65], ali za širu upotrebu nema opravdanja, jer neuronske mreže sa dva skrivena sloja mogu uspešno da reše svaki problem koji je rešiv pomoću neuronskih mreža. Sa druge strane, vreme potrebno za obuku se značajno povećava sa brojem skrivenih slojeva [66]. Imajući sve ovo u vidu, pri strukturisanju neuronske mreže za rešavanje relativno komplikovanih problema nameće se samo dilema da li izabрати topologiju sa jednim ili sa dva skrivena sloja. Ova dilema se najčešće razrešava izborom jednog skrivenog sloja. Ako se time ne postignu potpuno zadovoljavajući rezultati ili se pojave problemi sa memorijskim resursima računara zbog prevelikog broja neurona u tom jednom skrivenom sloju, onda se generiše neuronska mreža sa dva skrivena sloja i sa manjim ukupnim brojem skrivenih neurona u odnosu na prvi slučaj.

Što se tiče aktivacione funkcije neurona, situacija je prilično jasna. Kod višeslojnog perceptrona je bitno da ta funkcija bude diferencijabilna, zbog algoritma obuke, o kojem će kasnije biti više reči. Najviše se primenjuju identička (linearna) i sigmoidne funkcije, koje su prikazane u *Tab. 3.1*. Linearna aktivaciona funkcija se po pravilu koristi u neuronima ulaznog sloja. U neuronima skrivenih slojeva aktivacione funkcije su sigmoidne, čime neuronska mreža dobija svojstvo nelinearnosti, što joj daje veliku moć modelovanja veoma složenih realnih problema. Za neurone izlaznog sloja karakteristična je ili sigmoidna ili linearna funkcija, u zavisnosti od vrste zadatka koji se rešava neuronskom mrežom. Ukoliko je u pitanju klasifikovanje uzoraka, tada izlazne veličine neuronske mreže uglavnom imaju male vrednosti, pa je pogodno da aktivaciona funkcija izlaznog sloja bude oblika neke funkcije zasićenja, kao što su sigmoidne funkcije. Ako neuronska mreža rešava problem aproksimacije funkcije ili bilo koji drugi problem u kojem izlazne veličine mogu imati proizvoljne vrednosti, onda sigmoidne funkcije nisu pogodne za primenu kao aktivacione funkcije izlaznih neurona, pa se u tu svrhu koristi linearna funkcija.

### 3.4 Proces obuke neuronske mreže

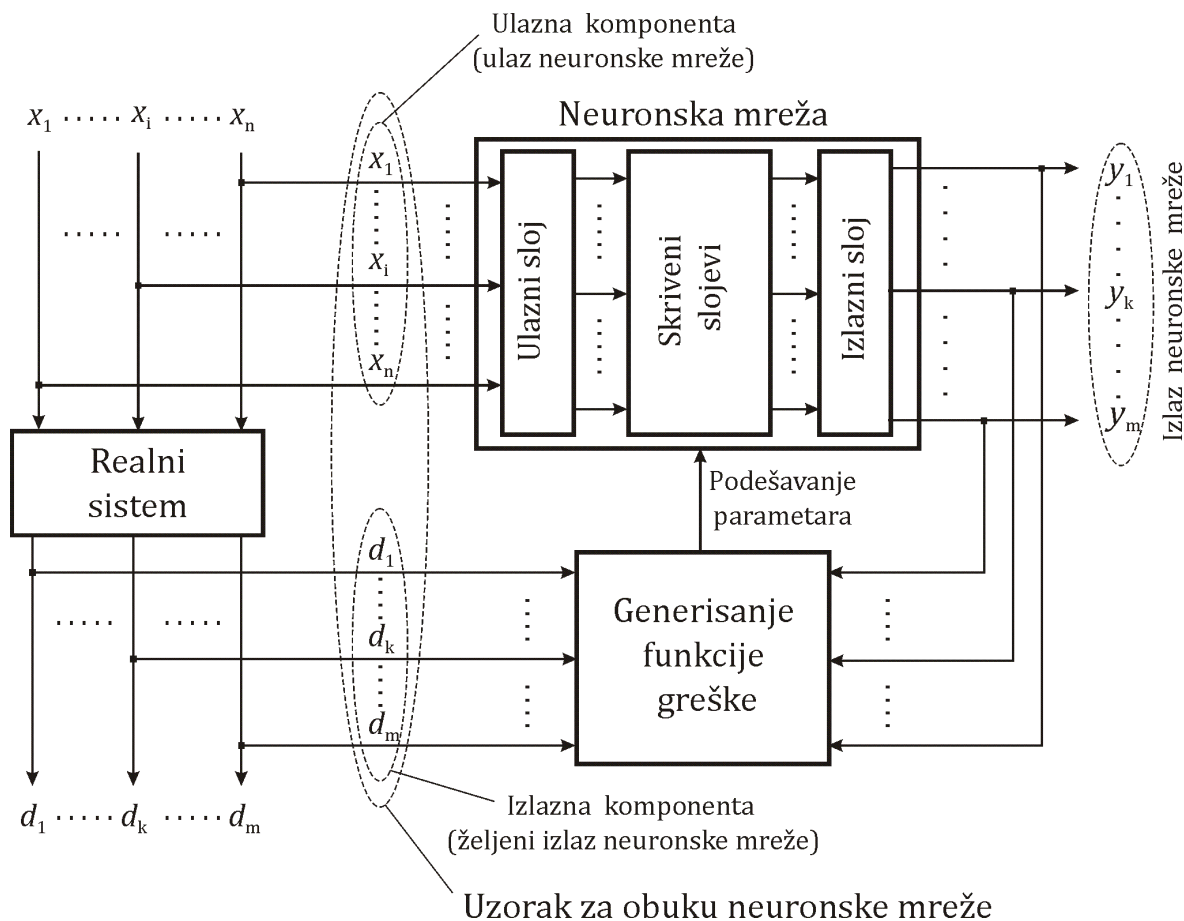
Posle strukturisanja neuronske mreže, tj. izbora njene topologije i aktivacionih funkcija pojedinih slojeva neurona, neophodno je da se sprovede proces učenja, kako bi mreža mogla da se koristi. U procesu učenja neuronska mreža stiče znanje o sistemu koji se pomoću nje modelira. S obzirom da se znanje čuva u težinskim koeficijentima sinaptičkih veza između neurona, proces učenja predstavlja podešavanje vrednosti tih težinskih koeficijenata u skladu sa nekim pravilom učenja, tako da neuronska mreža što bolje oponaša sistem koji se modeluje. Cilj je da se neuronska mreža obuči tako, da se postigne balans između njene sposobnosti da ispravno „odgovori“ na ulazne podatke koji su korišćeni u obuci i sposobnosti da generiše prihvatljive izlaze za one ulazne podatke koji nisu korišćeni prilikom obuke.

Proces učenja, koji se primenjuje kod višeslojnih neuronskih mreža bez povratne sprege, spada u tzv. nadgledano učenje<sup>15</sup> [67]. Pri nadgledanom učenju, neuronskoj mreži se dovodi niz uzoraka za obuku, pri čemu svaki uzorak za obuku ima svoju ulaznu i izlaznu komponentu, kako je prikazano na *Sl. 3.4*. Uzorci za obuku sadrže podatke dobijene

---

<sup>15</sup> eng. *supervised learning*

eksperimentalnim određivanjem ulaznih ( $x_1, \dots, x_n$ ) i izlaznih veličina ( $d_1, \dots, d_m$ ) realnog sistema, koji se želi modelovati neuronskom mrežom. Vrednosti ulaznih veličina realnog sistema čine ulaznu, a vrednosti njegovih izlaznih veličina čine izlaznu komponentu uzorka za obuku. Ulazna komponenta uzorka za obuku može se posmatrati kao ulaz neuronske mreže, a izlazna komponenta kao željeni izlaz neuronske mreže, na nivou uzorka. Svaki uzorak za obuku predstavlja jedan primer kojim se neuronskoj mreži predočava kako funkcioniše sistem koji treba da se modeluje. Veliki broj takvih primera treba da omogući da mreža stekne dovoljno znanja o sistemu, jer neuronske mreže upravo i uče kroz primere. Nadgledano učenje je jedan iterativni postupak, koji se zasniva na poređenju izlaza ( $y_1, \dots, y_m$ ) neuronske mreže sa izlaznom komponentom ( $d_1, \dots, d_m$ ) uzorka za obuku. Ovde se pod izlazom neuronske mreže podrazumeva rezultat koji se dobija na izlaznom sloju neuronske mreže, posle obrade ulazne komponente jednog uzorka za obuku (Sl. 3.4). Usled razlike između izlaza neuronske mreže i izlazne komponente odgovarajućeg uzorka za obuku, generiše se neka funkcija greške. Ova funkcija se koristi za preračunavanje sinaptičkih težinskih koeficijenata (podešavanje parametara), sa ciljem da se ta greška smanji, a time i poboljšaju performanse neuronske mreže.



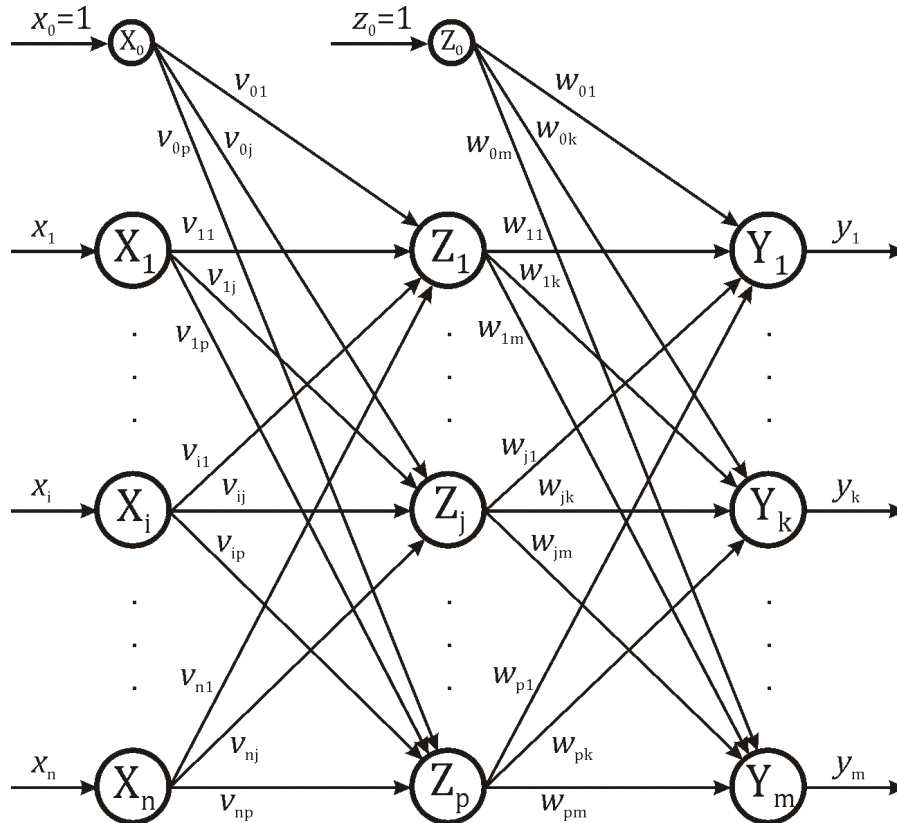
Sl. 3.4 Šematski prikaz nadgledanog učenja neuronske mreže

U nadgledanom učenju neuronskih mreža bez povratnih sprega, za optimizaciju funkcije greške može se koristiti bilo koja standardna numerička metoda. Međutim, najbolje su se pokazale metode u kojima se izračunavaju gradijent ili Jakobijan funkcije greške po težinskim koeficijentima [68]. Najpopularniji algoritam za izračunavanje i gradijenta i Jakobijana je algoritam sa prostiranjem greške unazad (eng. *error back-propagation* algoritam ili skraćeno BP algoritam). Zbog kompaktnosti, u daljem tekstu će se ovaj algoritam zvati BP algoritmom. Čak se i višeslojni perceptron često naziva BP neuronskom

mrežom, jer je u procesu njegovog učenja dominantna primena BP algoritma i nekih njegovih varijacija.

### 3.4.1 BP algoritam – teorijske osnove

Za početak masovne primene BP algoritma zaslužni su Rumelhart, Hinton i Williams [69], a njegovo detaljno izvođenje može se naći u mnogim referencama koje se bave neuronskim mrežama, kao što su [53], [54] i [62]. Ovde će najpre biti objašnjena suština BP algoritma, a zatim će biti opisano kako teče obuka neuronske mreže primenom ovog algoritma, korak po korak. U tu svrhu biće korišćena šema neuronske mreže sa jednim skrivenim slojem, data na Sl. 3.5.



Sl. 3.5 Šema BP neuronske mreže sa jednim skrivenim slojem

Na prikazanoj neuronskoj mreži ulazni neuroni su obeleženi sa  $X_i$  ( $i=1, \dots, n$ ), skriveni neuroni sa  $Z_j$  ( $j=1, \dots, p$ ) i izlazni neuroni sa  $Y_k$  ( $k=1, \dots, m$ ). Sinaptički težinski koeficijenti između ulaznog i skrivenog sloja obeleženi su oznakama  $v_{ij}$ , a između skrivenog i izlaznog sloja oznakama  $w_{jk}$ . Izlazni i skriveni neuroni mogu imati i pragove. Prag proizvoljnog izlaznog neurona  $Y_k$  označen je sa  $w_{0k}$ , a prag proizvoljnog skrivenog neurona  $Z_j$  označen je sa  $v_{0j}$ . U vezi sa time na slici su ucrtani fiktivni neuroni  $X_0$  i  $Z_0$  sa ulazima  $x_0 = 1$  i  $z_0 = 1$ , kako bi se dobili jednostavniji izrazi u jednačinama koje slede.

Neka skup podataka za obuku neuronske mreže sadrži  $n_{uz}$  uzoraka za obuku. Tada se ulazna komponenta proizvoljnog uzorka može obeležiti sa  $(x_{1s}, \dots, x_{ns})$ , a odgovarajuća izlazna komponenta sa  $(d_{1s}, \dots, d_{ms})$ , pri čemu ( $s=1, \dots, n_{uz}$ ). Međutim, zbog jednostavnosti, nastavak objašnjenja odnosiće se na jedan uzorak, a kasnije će u opisu funkcionisanja algoritma biti uvedeno proširenje koje se odnosi na ceo skup podataka za obuku. Zato će, za sada, indeks  $s$  biti izostavljen, pa će ulazna i izlazna komponenta uzorka za obuku biti označene sa  $(x_1, \dots, x_n)$  i  $(d_1, \dots, d_m)$ , kako je urađeno i na Sl. 3.4.

U jednom prolazu kroz neuronsku mrežu od ulaznih ka izlaznim neuronima, svaki ulazni neuron  $X_i$  ( $i=1, \dots, n$ ) preuzima svoj ulazni signal  $x_i$  i šalje ga prema svakom skrivenom neuronu  $Z_j$  ( $j=1, \dots, p$ ), kako pokazuje Sl. 3.5. Po ugledu na jednačine (3.2) i (3.3), svaki skriveni neuron izračunava svoj izlazni signal  $z_j$  primenom aktivacione funkcije  $f_z$  na sumu ponderisanih ulaznih signala,

$$z_j = f_z \left( \sum_{i=0}^n (x_i v_{ij}) \right), \quad j = 1, \dots, p \quad (3.4)$$

i šalje taj signal svim izlaznim neuronima. Zatim svaki izlazni neuron  $Y_k$  ( $k=1, \dots, m$ ) na sličan način, primenom aktivacione funkcije  $f_y$  izračunava svoj izlazni signal

$$y_k = f_y \left( \sum_{j=0}^p (z_j w_{jk}) \right), \quad k = 1, \dots, m. \quad (3.5)$$

Određivanjem izlaza neuronske mreže ( $y_1, \dots, y_m$ ) završava se prolaz unapred, posle čega sledi prolaz unazad.

Prolaz unazad počinje poređenjem dobijenog izlaza neuronske mreže ( $y_1, \dots, y_m$ ) i izlazne komponente uzorka za obuku ( $d_1, \dots, d_m$ ), na svakom izlaznom neuronu, pri čemu se generišu greške  $e_k = d_k - y_k$ , ( $k = 1, \dots, m$ ). Ovo su pojedinačne greške na izlaznim neuronima, ali da bi se ocenila greška neuronske mreže na nivou uzorka, formira se funkcija greške, najčešće u obliku ukupne kvadratne greške, prema izrazu

$$E = \frac{1}{2} \sum_{k=1}^m (d_k - y_k)^2 \quad (3.6)$$

ili srednje kvadratne greške,

$$E = \frac{1}{2 \cdot m} \sum_{k=1}^m (d_k - y_k)^2.$$

Uvrštavanjem (3.5) u (3.6) dobija se

$$E = \frac{1}{2} \sum_{k=1}^m \left( d_k - f_y \left( \sum_{j=0}^p (z_j w_{jk}) \right) \right)^2, \quad (3.7)$$

čime se pokazuje da funkcija greške  $E$  zavisi od težinskih koeficijenata sinapsi između skrivenog i izlaznog sloja, kao i od aktivacija skrivenih neurona, a to se skraćeno može napisati kao

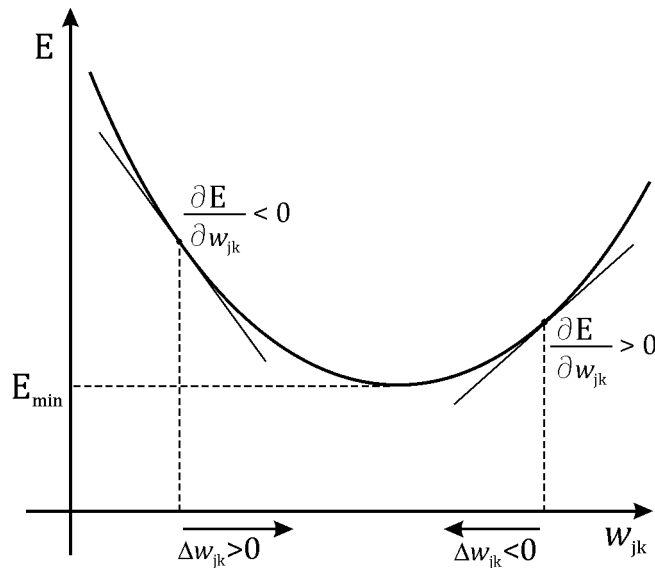
$$E = f(w_{jk}, z_j), \quad (j = 0, \dots, p), \quad (k = 1, \dots, m). \quad (3.8)$$

Ovo je veoma bitno, jer se pomoću parcijalnih izvoda funkcije greške po svakom težinskom koeficijentu može utvrditi, u kom smeru treba korigovati konkretni težinski koeficijent da bi se funkcija greške smanjila, što je ilustrovano na Sl. 3.6. Na slici se vidi da u oblastima gde je funkcija greške monotono opadajuća ( $\frac{\partial E}{\partial w_{jk}} < 0$ ) treba povećati težinski koeficijent ( $\Delta w_{jk} > 0$ ), a tamo gde je rastuća ( $\frac{\partial E}{\partial w_{jk}} > 0$ ), treba ga smanjiti ( $\Delta w_{jk} < 0$ ), da bi se funkcija greške približavala vrednosti  $E_{\min}$ .

Po metodi opadajućeg gradijenta, korekcija težinskog koeficijenta određuje se prema izrazu [62]

$$\Delta w_{jk} = -\gamma \frac{\partial E}{\partial w_{jk}}, \quad (j = 0, \dots, p), \quad (k = 1, \dots, m), \quad (3.9)$$

gde je  $\gamma$  koeficijent brzine učenja, kojim se reguliše veličina korekcije u jednoj iteraciji. Za koeficijent  $\gamma$  obično se koriste vrednosti između 0 i 1. Vrednosti bliže nuli označavaju sporije učenje, zbog većeg broja potrebnih iteracija da se dostigne vrednost  $E_{\min}$ . Vrednosti bliže jedinici imaju za posledicu brže učenje, ali može da se desi i da, zbog velikog inkrementa  $\Delta w_{jk}$ , vrednost  $E$  osciluje oko  $E_{\min}$  i da je nikada ne dostigne.



Sl. 3.6 Uticaj promene težinskog koeficijenta na funkciju greške

Diferenciranjem jednačine (3.7) i primenom pravila diferenciranja složenih funkcija, dolazi se do izraza

$$\Delta w_{jk} = \gamma \cdot \delta_k \cdot z_j, \quad (j = 0, \dots, p), \quad (k = 1, \dots, m), \quad (3.10)$$

gde je  $\delta_k$  signal greške, koji se dobija iz

$$\delta_k = (d_k - y_k) \cdot f'_y \left( \sum_{j=0}^p (z_j w_{jk}) \right), \quad (k = 1, \dots, m). \quad (3.11)$$

Na sličan način dobijaju se i izrazi za određivanje korekcije težinskih koeficijenata sinapsi između ulaznog i skrivenog sloja

$$\Delta v_{ij} = \gamma \cdot \delta_j \cdot x_i, \quad (i = 0, \dots, n), \quad (j = 1, \dots, p), \quad (3.12)$$

i odgovarajućih signala greške

$$\delta_j = \sum_{k=1}^m (\delta_k w_{jk}) \cdot f'_z \left( \sum_{i=0}^n (x_i v_{ij}) \right), \quad (j = 0, \dots, p). \quad (3.13)$$

Posmatrajući jednačine (3.11) i (3.13), vidi se da samo po jednom uzorku, treba izračunati prve izvode aktivacionih funkcija,  $f'_y$  i  $f'_z$ , ukupno  $(m + p)$  puta. Imajući u vidu da taj broj treba pomnožiti sa brojem uzoraka za obuku, pa onda još i sa ukupnim brojem iteracija u toku obuke, moglo bi se pretpostaviti da je određivanje vrednosti signala greške veoma zahtevan proces sa stanovišta računarskih i vremenskih resursa. Međutim, pošto su aktivacione funkcije kod BP mreža isključivo linearne ili sigmoidne, ovaj problem je mnogo manji, nego što izgleda. O izvodu linearne funkcije ne treba posebno govoriti, ali kad su u pitanju sigmoidne funkcije, treba pomenuti jednu njihovu veoma korisnu karakteristiku. Naime, izvodi sigmoidnih funkcija mogu se izraziti pomoću samih tih funkcija korišćenjem dve ili tri aritmetičke operacije, tako da u računarskoj implementaciji BP algoritma uopšte nema potrebe za diferenciranjem.

Posle određivanja korekcija težinskih koeficijenata, ostaje još samo da se preračunaju sami težinski koeficijenti. To se izvodi korišćenjem jednačina

$$w_{jk}(\text{novi}) = w_{jk}(\text{stari}) + \Delta w_{jk}, \quad (j = 0, \dots, p), \quad (k = 1, \dots, m) \quad (3.14)$$

i

$$v_{ij}(\text{novi}) = v_{ij}(\text{stari}) + \Delta v_{ij}, \quad (i = 0, \dots, n), \quad (j = 1, \dots, p). \quad (3.15)$$

Jednačine (3.10) - (3.15) koriste se pri prolazu kroz neuronsku mrežu unazad (od izlaznih ka ulaznim neuronima) i predstavljaju suštinu BP algoritma. Njihovo kompletno izvođenje može se pronaći u referencama [53], [54] i [62].

U nastavku se, korak po korak, opisuje način na koji se neuronska mreža obučava pomoću BP algoritma.

### 3.4.2 Obuka neuronske mreže korišćenjem BP algoritma

Ukoliko je neuronskoj mreži na raspolaganju skup podataka za obuku koji sadrži  $n_{uz}$  uzoraka, proces obuke se odvija kroz sledeće korake.

#### Korak 0. Inicijalizacija

- Usvaja se vrednost za koeficijent učenja  $\gamma$ .
- Postavlja se kumulativna vrednost funkcije greške na nulu, tj.  $E = 0$ .
- Usvaja se maksimalno dozvoljena vrednost funkcije greške  $E_{max}$ .
- Postavljaju se početne vrednosti za sinaptičke težinske koeficijente  $v_{ij}$  ( $i=0, \dots, n$ ;  $j=1, \dots, p$ ) i  $w_{jk}$  ( $j=0, \dots, p$ ;  $k=1, \dots, m$ ). Obično su to male vrednosti iz intervala  $[-0.5, 0.5]$  ili  $[-1, 1]$ , generisane po principu slučajnih brojeva.
- Usvaja se maksimalni željeni broj ciklusa učenja  $q_{max}$ .
- Postavlja se početna vrednost brojača ciklusa učenja na jedinicu,  $q = 1$ .
- Postavlja se početna vrednost brojača uzoraka za obuku na jedinicu,  $s = 1$ .

#### Korak 1. Prosleđivanje neuronskoj mreži $s$ -tog uzorka za obuku

Neuronska mreža prihvata  $s$ -ti uzorak za obuku, sastavljen od ulazne komponente  $(x_{1s}, \dots, x_{ns})$  i izlazne komponente  $(d_{1s}, \dots, d_{ms})$ ,

#### Korak 2. Obrada ulazne komponente $s$ -tog uzorka za obuku

Ulazna komponenta  $s$ -tog uzorka,  $(x_{1s}, \dots, x_{ns})$ , obrađuje se u prolazu unapred, od ulaznih ka izlaznim neuronima, primenom jednačine (3.4) na svakom neuronu skrivenog sloja, pa zatim primenom jednačine (3.5) na svakom izlaznom neuronu. Kao rezultat, dobija se izlaz neuronske mreže  $(y_{1s}, \dots, y_{ms})$ .

#### Korak 3. Izračunavanje funkcije greške

- Određuje se funkcija greške na nivou  $s$ -tog uzorka  $E_s$ , prema jednačini (3.6),
- Određuje se kumulativna vrednost funkcije greške za sve uzorke obrađene do ovog momenta, prema izrazu:  $E \leftarrow E + E_s$ .

#### Korak 4. Izračunavanje signala greške izlaznog sloja neurona

Izračunavaju se signali greške  $\delta_k$ , ( $k = 1, \dots, m$ ), za svaki neuron u izlaznom sloju, prema jednačini (3.11).



**Korak 5. Podešavanje parametara izlaznog sloja neurona**

Za  $(j = 0, \dots, p)$  i  $(k = 1, \dots, m)$  izračunavaju se iznosi korekcije,  $\Delta w_{jk}$ , i korigovane vrednosti težinskih koeficijenata,  $w_{jk}$ , svih sinapsi koje vode do izlaznih neurona, prema jednačinama (3.10) i (3.14), respektivno.

**Korak 6. Izračunavanje signala greške skrivenog sloja neurona**

Izračunavaju se signali greške  $\delta_j$ ,  $(j = 0, \dots, p)$ , za svaki neuron u skrivenom sloju, prema jednačini (3.13).

**Korak 7. Podešavanje parametara skrivenog sloja neurona**

Za  $(i = 0, \dots, n)$ ,  $(j = 1, \dots, p)$  izračunavaju se iznosi korekcije  $\Delta v_{ij}$  i korigovane vrednosti  $v_{ij}$  težinskih koeficijenata svih sinapsi koje povezuju ulazni i skriveni sloj, prema jednačinama (3.12) i (3.15), respektivno.

**Korak 8. Provera da li je obrađen ceo skup podataka za obuku**

Proverava se da li ima još uzoraka u skupu za obuku. To se radi poređenjem brojača uzoraka  $s$  sa ukupnim brojem uzoraka u skupu za obuku  $n_{uz}$ , tj. proverava se da li je  $s = n_{uz}$ . Ako nije ispunjen ovaj uslov, onda se brojač povećava za jedan ( $s = s + 1$ ) i procedura se vraća na Korak 1. Na taj način, prelazi se na sledeći uzorak, pa se ponavljaju Koraci 1-8. Kada se dođe do poslednjeg uzorka u skupu za obuku, prelazi se na Korak 9.

**Korak 9. Provera uslova za završetak i prekid obuke**

Proverava se da li je kumulativna vrednost funkcije greške  $E$  manja od maksimalno dozvoljene vrednosti  $E_{max}$ . Ako jeste, obuka je završena.

Ako nije ispunjen uslov za završetak obuke, onda se proverava da li su ispunjeni uslovi za prekid obuke. Do prekida obuke može doći u nekom od sledećih slučajeva:

- dostignut je maksimalni željeni broj ciklusa učenja  $q = q_{max}$  ili
- došlo je do zasićenja pri smanjivanju funkcije greške, tj. dalja obuka ne dovodi do poboljšanja performansi ili
- prošlo je maksimalno vreme predviđeno za obuku.

Ako nije ispunjen ni jedan uslov za prekid obuke, onda se obuka nastavlja, tako što počinje sledeći ciklus učenja. To se ostvaruje na sledeći način:

- povećava se brojač ciklusa učenja za jedan,  $q = q + 1$ ,
- vrednost kumulativne greške se postavlja na nulu,  $E = 0$ ,
- vrednost brojača uzoraka se postavlja na 1,  $s = 1$  i
- procedura se vraća na Korak 1.

Treba napomenuti da sledeći ciklus učenja počinje sa parametrima neuronske mreže koji su izračunati u prethodnom ciklusu učenja, kako bi se očuvao kontinuitet učenja kroz cikluse.

Na opisani način BP algoritam omogućava sprovođenje procesa obuke neuronske mreže sa jednim skrivenim slojem neurona. Ukoliko bi skrivenih slojeva bilo više, onda bi ova procedura bila neznatno modifikovana, u smislu da bi se obrada uzorka za obuku u Koraku 2 izvršila onoliko puta, koliko ima skrivenih slojeva neurona. Isto tako, Koraci 6 i 7 bi se morali izvršiti po jednom za svaki skriveni sloj neurona.

Način obuke neuronske mreže, predstavljen u ovom odeljku, je tzv. individualni način obuke<sup>16</sup>, gde se korekcija težinskih koeficijenata izvodi nakon obrade svakog uzorka za obuku. Postoji i grupni način obuke<sup>17</sup>, sa podešavanjem težinskih koeficijenata nakon obrade svih uzoraka iz skupa podataka za obuku. Generalno, do danas su razvijene mnoge varijacije BP algoritma, sa namerom da se poboljša postupak obuke. U tom smislu su u osnovni BP algoritam ugrađivane npr. tehnike za izbegavanje zaglavlivanja u lokalnim minimumima funkcije greške ili tehnike koje omogućavaju promenljivost koeficijenta brzine učenja u toku obuke itd. Ove novije verzije uglavnom dovode do ubrzanja obuke, ali su kompleksnije i zahtevnije prema memorijskim resursima računara. Detalji o unapređenim verzijama BP algoritma, kao što su Levenberg-Marquardt-ov algoritam, konjugovano-gradijentni BP algoritam, skalirani konjugovano-gradijentni BP algoritam, elastični BP algoritam itd. prevazilaze okvire teme ovog istraživanja i mogu se naći u referencama [53], [54], [62], [68] i [70]. Ovde je bilo bitno da se razjasne osnove procesa obuke neuronskih mreža posredstvom BP algoritma, a te osnove su iste, bez obzira na razlike koje postoje među pojedinim verzijama ovog algoritma.

U današnje vreme, svi algoritmi za obuku neuronskih mreža implementirani su u alatu Neural Network Toolbox koji pripada softverskom paketu Matlab, još od njegovih najranijih verzija. Zbog toga istraživač, koji želi da realizuje neku konkretnu neuronsku mrežu, nema potrebe da programira procedure za implementaciju ovih algoritama, ali može izrađivati programe koji će koristiti gotove algoritme u procesu realizacije mreže. Međutim, moraju se poznavati parametri neuronske mreže, čijim se variranjem mogu menjati njene performanse. Zatim, treba znati osnovne karakteristike pojedinih algoritama za obuku, kako bi se mogao izabrati onaj, koji je najpogodniji za rešavanje konkretnog zadatka. Dalje, treba znati svojstva aktivacionih funkcija, jer se različitim aktivacionim funkcijama dobijaju veoma različiti rezultati u primeni neuronskih mreža. Svi ovi parametri se mogu varirati, dok se ne dođe do zadovoljavajućeg rešenja. U nastavku se daje jedna okvirna procedura koja opisuje put od ideje do realizacije neuronske mreže, namenjene za rešavanje zadataka prepoznavanja uzoraka i njihovog razvrstavanja u neku od unapred definisanih klasa, što najpribližnije odgovara dijagnostičkim zadacima postavljenim na početku ovog rada.

### 3.5 Postupak razvoja neuronske mreže za prepoznavanje uzoraka

Dijagram redosleda faza u procesu razvoja jedne opšte neuronske mreže prikazan je na Sl. 3.7, a u nastavku slede određene preporuke za svaku od faza, sa naglaskom na problem prepoznavanja uzoraka, koji je ovde od posebnog interesa.

#### Faza 1. Definisane ulaznih i izlaznih veličina

Najbitnija stvar kod definisanja ulaznih i izlaznih veličina neuronske mreže jeste njihova dostupnost. Drugim rečima, sa stanovišta neuronskih mreža, neka veličina može biti ulazna ili izlazna samo ukoliko se njena vrednost može odrediti i biti na raspolaganju onome, ko kreira neuronsku mrežu.

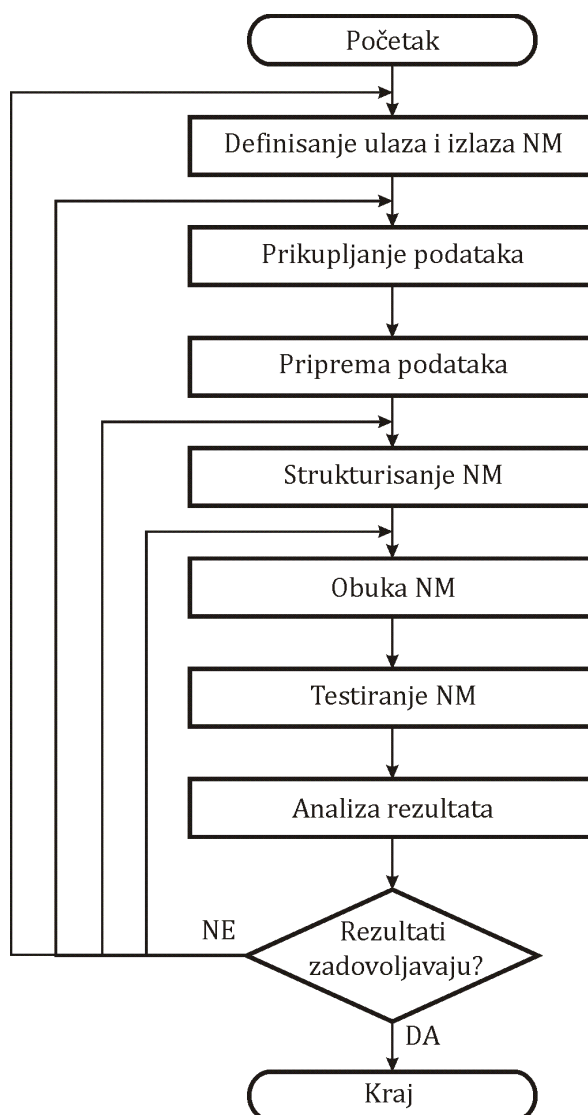
Kad je u pitanju prepoznavanje uzoraka, zadatak određivanja broja izlaznih veličina je trivijalan, jer se, u najvećem broju slučajeva, taj broj poklapa sa brojem predefinisanih klasa u koje treba razvrstati uzorke, pri čemu svaka izlazna veličina reprezentuje jednu klasu. Broj ulaznih veličina nije tako lako definisati. Bitno je da to bude minimalan broj koji

---

<sup>16</sup> eng. *incremental training*

<sup>17</sup> eng. *batch training*

će na zadovoljavajući način opisati realni sistem na koji se klasifikacija odnosi. To znači da treba dobro poznavati realni sistem i uzajamne zavisnosti veličina koje opisuju njegovo ponašanje.



Sl. 3.7 Faze razvoja neuronske mreže

## Faza 2. Prikupljanje podataka za obuku

Podaci za obuku se dobijaju eksperimentalnim putem, bilo da je u pitanju realni ili virtuelni eksperiment. S obzirom da višeslojni perceptroni imaju sposobnost dobre interpolacije, ali ne i ekstrapolacije [71], veoma je važno da skup podataka za obuku dobro pokrije ceo opseg vrednosti koje ulazne veličine mogu da imaju. Kada je u pitanju zadatak prepoznavanja uzoraka, pomenuti uslov bi trebalo da bude ispunjen u okviru svake klase. Osim toga, preporučljivo je da uzorci za obuku sadrže što je moguće širi opseg varijacija ulaznih veličina ili čak i uticaj šuma na njihove izmerene vrednosti.

## Faza 3. Priprema podataka za obuku

U cilju poboljšanja efikasnosti obuke, neophodno je da se izvrši priprema podataka za obuku. Prvi deo te pripreme čini pred-obrađivanje podataka, a drugi deo je podela skupa za obuku na odgovarajuće podskupove.

Pred-obrada podataka je neophodna zbog prirode sigmoidnih aktivacionih funkcija koje se dominantno koriste kod višeslojnog perceptrona. Pošto se vrednosti ovih funkcija kreću u intervalu  $(-1,1)$  ili  $(0,1)$ , i veoma brzo dostižu zasićenje (Tab. 3.1), nije poželjno da opsezi vrednosti promenljivih iz uzorka za obuku imaju veliki raspon. Zbog toga se praktikuje da se sve promenljive normalizuju, tj. skaliraju tako da i njihove vrednosti budu koncentrisane u intervalu  $(-1,1)$  ili  $(0,1)$ , u skladu sa aktivacionim funkcijama. Normalizacija se može vršiti na različite načine. Ovde se, samo radi ilustracije, daje jedan od najčešćih načina za normalizaciju promenljive  $X$  unutar intervala između bilo koja dva realna broja  $a$  i  $b$ :

$$X^* = a + (b - a) \cdot \frac{X - X_{\min}}{X_{\max} - X_{\min}}$$

gde su  $X^*$ ,  $X_{\max}$  i  $X_{\min}$  normalizovana, maksimalna i minimalna vrednost, respektivno. Inače, normalizaciji podleže kompletni uzorak za obuku, i ulazna i izlazna komponenta.

Drugi deo pripreme podataka za obuku je njihova podela na tri podskupa: skup za obuku, skup za validaciju i skup za testiranje. Prvi od njih se koristi za izračunavanje korigovanih vrednosti težinskih koeficijenata. Skup podataka za validaciju se koristi za određivanje minimalne greške validacije, čijim dostizanjem se obuka prekida i sprečava se „pretreniranost“ neuronske mreže. „Pretrenirana“ neuronska mreža odlično klasifikuje uzorke koji su korišćeni za obuku, ali loše klasifikuje nove uzorke, što znači da loše generalizuje. Ovde je možda zgodno napomenuti da problem „pretreniranosti“ neuronske mreže može da se izbegne, ako se obezbedi dovoljno veliki broj uzoraka za obuku. Treći podskup podataka (podaci za testiranje) koristi se za ocenu performansi modela. Preporučeni odnos ova tri podskupa je 60% podataka za obuku, 20% za validaciju i 20% za testiranje, ali može biti podložan promenama [71]. Podaci se u podskupove razvrstavaju najčešće na principu slučajnih brojeva, ali se može izvršiti i kontrolisano formiranje tih podskupova.

#### Faza 4. Strukturisanje neuronske mreže

U okviru strukturisanja neuronske mreže treba izvršiti nekoliko zadataka:

- odabrati vrstu neuronske mreže,
- odrediti topologiju neuronske mreže,
- izabrati aktivacione funkcije po slojevima neurona,
- izabrati konkretan algoritam za obuku,
- definisati merilo za ocenjivanje neuronske mreže

Izbor vrste neuronske mreže zavisi od problema koji se želi rešiti, pa se uglavnom biraju one koje su se najbolje pokazale, pri rešavanju date vrste problema. Tako, na primer, kada se govori o prepoznavanju uzoraka kod kojih nije bitan faktor vremena, trebalo bi izabrati neku neuronsku mrežu bez povratnih sprega. Ako bi trebalo uzeti u obzir i vreme, onda se mora koristiti neka rekurentna neuronska mreža (sa povratnim spregama).

O određivanju topologije neuronske mreže bilo je dosta reči u odeljku 3.3. Ukratko, treba pokušati rešiti problem sa jednim skrivenim slojem neurona, a ako rezultati nisu zadovoljavajući, onda pridodati još jedan skriveni sloj, ali se dalje povećanje ovog broja ne preporučuje. Najpouzdaniji način za određivanje broja skrivenih neurona u jednom sloju jeste metoda probe i greške.

Izbor aktivacionih funkcija po slojevima neurona zavisi od izabrane vrste neuronske mreže, ali i od problema koji se rešava. O tome je bilo reči u odeljku 3.2. Ovde treba još dodati da i raspon vrednosti podataka za obuku utiče na izbor aktivacione funkcije, naročito u izlaznom sloju neurona. Konkretno, kod problema prepoznavanja uzoraka, poželjno je da izlazi neuronske mreže budu u nekom uskom intervalu vrednosti. To znači da bi za izlazni sloj bila pogodna neka sigmoidna aktivaciona funkcija, dok linearna ne bi dala zadovoljavajuće rezultate.

Često je već samim izborom vrste neuronske mreže definisan i konkretan algoritam koji će se koristiti za njenu obuku ili, ako ne konkretan algoritam, onda grupa algoritama koja se primenjuje za obuku izabrane vrste neuronske mreže. U slučaju postojanja grupe algoritama koji su u užem izboru za obuku, treba izabrati onaj, koji se najbolje pokazao u rešavanju problema, sličnih datom problemu. Pošto je u ovom poglavlju fokus usmeren na BP neuronske mreže, ovde treba izabrati neku od verzija BP algoritma. Najbrži od njih je Levenberg-Marquardt-ov algoritam, ali zahteva značajno angažovanje memorijskih resursa računara, pa nije pogodan za velike neuronske mreže sa nekoliko hiljada težinskih koeficijenata. Osim toga, bolje se ponaša u rešavanju problema aproksimacije funkcija, nego kod prepoznavanja uzoraka. Kao dobar izbor za obuku velikih neuronskih mreža i kod zadataka prepoznavanja uzoraka, pokazali su se konjugovano-gradijentni BP algoritam i elastični BP algoritam. Ovi algoritmi nisu previše zahtevni u memorijskom smislu, a mnogo su brži od standardnog BP algoritma.

Inače, proces izbora algoritma za obuku neuronske mreže izvodi se kroz sam proces obuke, isprobavanjem različitih algoritama iz užeg izbora i praćenjem rezultata koji se postižu pomoću svakog konkretnog algoritma. Zbog toga se za izbor algoritma ne može sa sigurnošću reći da li pripada fazi strukturisanja neuronske mreže, ili fazi obuke, ili obema fazama u razvoju neuronske mreže. Generalno, to i nije mnogo važno, jer se postupak razvoja neuronske mreže odvija u stalnim iteracijama kroz razne faze, sve dok se ne dođe do neuronske mreže koja daje zadovoljavajuće rezultate, kako je prikazano na *Sl. 3.7*.

Po pitanju definisanja merila za ocenjivanje neuronske mreže, moglo bi se reći da ne postoji neko opšte pravilo, po kojem se neka veličina greške kod neuronskih mreža može smatrati minimalnom. U suštini, uspešno obučenom neuronskom mrežom može se smatrati ona koja zadovoljava potrebe korisnika. Konkretno, kod problema prepoznavanja uzoraka i njihovog razvrstavanja u predefinisane klase, kao merilo bi mogao da se uzme udeo (ili procenat) dobro klasifikovanih uzoraka u odnosu na ukupan broj uzoraka.

### Faza 5. Obuka neuronske mreže

Suština procesa obuke je detaljno opisana u odeljku 3.4.2, a sa stanovišta projektanta neuronske mreže bitno je usvojiti vrednosti određenih parametara, kako bi se obuka mogla izvršiti. Najpre treba usvojiti koeficijent brzine učenja, prema preporukama datim odmah posle jednačine (3.9), gde se prvi put pominje ovaj parametar. Osim toga, treba definisati funkciju greške. U slučaju obuke neuronskih mreža bez povratnih sprega, to je najčešće srednja kvadratna greška ili ukupna kvadratna greška neuronske mreže. Posle definisanja funkcije greške, treba usvojiti željenu vrednost kojoj će funkcija greške da konvergira tokom obuke, čime se definiše uslov za završetak obuke. Da bi se izbeglo beskonačno trajanje obuke, ukoliko se funkcija greške nikad ne spusti do željene vrednosti, trebalo bi usvojiti vrednosti još nekih promenljivih. To su vrednosti koje, ako se dostignu, dovode do prekida obuke. Za većinu varijanti BP algoritma, treba definisati :

- maksimalni broj ciklusa učenja,
- maksimalno vreme trajanja obuke,
- minimalnu vrednost gradijenta greške,
- dozvoljeni broj validacionih provera.

Ako vrednost gradijenta greške padne ispod minimalne, to označava da dalji tok obuke ne dovodi do daljeg smanjenja funkcije greške, što dovodi do prekida obuke.

Dozvoljeni broj validacionih provera je maksimalni broj uzastopnih ciklusa učenja tokom kojih funkcija greške, izračunata na podacima za validaciju, raste. Porast vrednosti funkcije greške na podacima za validaciju označava da je ostvarena maksimalna sposobnost neuronske mreže da generalizuje, što uzrokuje prekid obuke.

Po usvajanju svih navedenih parametara i njihovog prosleđivanja odgovarajućem softveru, izvršava se obuka prema izabranoj verziji BP algoritma.

### **Faza 6. Testiranje neuronske mreže**

Po završetku obuke vrši se testiranje neuronske mreže novim podacima, koji nisu korišćeni prilikom obuke. Trebalo bi da ovi podaci takođe budu dobijeni eksperimentalno, iz modelovanog realnog sistema i trebalo bi da budu unutar opsega, koji pokrivaju podaci, korišćeni za obuku. Format svakog uzorka iz skupa podataka za testiranje mora odgovarati formatu ulazne komponente skupa podataka za obuku. Testiranje se obavlja tako što se jedan po jedan uzorak iz skupa za testiranje dovodi na ulaz obučene neuronske mreže. Neuronska mreža obrađuje te uzorke i daje izlazne rezultate. Konkretno, u slučaju problema klasifikacije uzoraka, dobijeni rezultati pokazuju kojoj predefinisanoj klasi pripada svaki od uzoraka iz skupa za testiranje. Proceduru testiranja neuronske mreže je poželjno ponoviti više puta sa različitim skupovima za testiranje, kako bi se mogao doneti što verodostojniji zaključak o kvalitetu obučene neuronske mreže.

### **Faza 7. Analiza dobijenih rezultata**

Određuje se procenat uspešnosti klasifikacije u okviru svakog skupa za testiranje. Ukoliko je taj procenat zadovoljavajući, postupak formiranja neuronske mreže se završava i ona se može dalje koristiti za klasifikaciju nekih novih uzoraka. Ako rezultati klasifikacije nisu zadovoljavajući, onda se treba vratiti na neku od prethodnih faza postupka formiranja neuronske mreže (Sl. 3.7).

Ono što prvo treba pokušati, ako su rezultati nezadovoljavajući, jeste ponavljanje procesa obuke neuronske mreže, bez prethodnog preduzimanja bilo kakvih promena. Naime, svaki put, kada se mreža obučava, čak i pri istim parametrima koje projektant mreže odabira, dobiće se različiti rezultati. Razlog je inicijalizacija sinaptičkih težinskih koeficijenata, koja se, kako je ranije rečeno, sprovodi po principu slučajnih brojeva. Isto tako, podela skupa za obuku na tri dela (obuka, validacija i testiranje) vrši se najčešće na principu slučajnih brojeva. Iz ovih razloga, može se desiti da, tokom procesa obuke, funkcija greške „upadne“ u lokalni minimum iz kojeg ne može da izađe, pa se dobijaju rezultati lošiji od očekivanih. Zato ponavljanje procesa obuke sa istim parametrima, čak i nekoliko puta, može da donese znatno bolje rezultate.

Ako višestruka ponovljena obuka ne dovede do poboljšanja performansi neuronske mreže, onda bi trebalo probati sa izmenama u strukturi mreže. Pokušati sa povećanjem broja neurona u skrivenom sloju ili sa dodavanjem novog skrivenog sloja. Takođe, možda bi bolji rezultati mogli da se dobiju, ako se izabere drugi algoritam za obuku ili ako se izmeni aktivaciona funkcija u nekom od slojeva. U krajnjem slučaju, pokušati sa nekom drugom vrstom neuronske mreže.

Ukoliko ni promene u strukturi neuronske mreže ne rezultiraju poboljšanjem performansi mreže, onda se treba vratiti još bliže početku kompletne procedure. Trebalo bi proveriti da li skup podataka za obuku ispunjava uslove navedene u opisu Faze 2. Ako postoji mogućnost, pokušati obogatiti skup podataka za obuku novim i raznovrsnijim uzorcima.

Konačno, ako ni dopuna skupa podataka za obuku ne doprinese poboljšanju performansi neuronske mreže, trebalo bi preispitati ulazne i izlazne veličine, koje možda nisu dobro izabrane i eventualno dodati neku novu ili izbaciti neku od postojećih.

## 4 OSREDNJENI MODEL MOTORA

Motor sa unutrašnjim sagorevanjem je komplikovan dinamički sistem, sastavljen od određenog broja podsistema, koji i sami mogu biti složeni, pa se matematičko modelovanje motora može smatrati veoma zahtevnim zadatkom. Zbog toga se, u zavisnosti od toga šta je cilj, predmet modelovanja sužava samo na određene procese koji se odvijaju u motoru i pri tome se uzimaju u obzir samo relevantni pod sistemi motora. Međutim, čak i u takvim slučajevima, modeli mogu da budu prilično složeni. Na primer, ukoliko je cilj modela da se detaljno i verno opišu termodinamički i kinetički procesi u toku sagorevanja, neophodni su veoma složeni višedimenzionalni modeli, koji sadrže veliki broj parcijalnih diferencijalnih jednačina i zahtevaju komplikovan matematički aparat za svoje rešavanje. Ovakvi modeli po pravilu daju rešenja visoke tačnosti, ali su i vrlo zahtevni sa aspekta utrošenog kompjuterskog vremena da bi se do tih rešenja došlo. Zbog toga se modeli tog nivoa složenosti koriste uglavnom u istraživanjima tokom projektovanja, razvoja i optimizacije vitalnih sklopova motora, ali nisu primenljivi u oblasti upravljanja motorom, gde je bitno da se jednačine iz modela rešavaju u realnom vremenu, tokom rada motora. Ova tvrdnja se još može potkrepiti činjenicom, da su čak i procesori elektronskih upravljačkih jedinica (ECU) najmodernijih motora mnogo sporiji nego procesori savremenih desktop računara. Razlog njihovih slabijih brzinskih performansi je taj, što moraju pouzdano da rade u okruženju u kojem vladaju teški uslovi rada npr. široki opseg temperatura od  $-40^{\circ}\text{C}$  do  $+100^{\circ}\text{C}$ , visoka vlažnost, prisustvo elektromagnetnih polja (sistema za paljenje, radio signala, ...). Zbog toga se mora racionalno upravljati resursima ovih procesora, a to znači da se u upravljanju motorima moraju primenjivati jednostavniji matematički modeli, koji se mogu implementirati u kontrolerima koji rade u realnom vremenu. Oni su u literaturi označeni kao modeli orijentisani ka upravljanju motorom ili **upravljački orijentisani modeli** (eng. *Control Oriented Models* - COM) [72]. Pošto upravljački orijentisani modeli motora rade u realnom vremenu, oni imaju potencijal da se iskoriste u dijagnostičke svrhe. Zato će u daljem tekstu biti malo više reči upravo o tim modelima.

### 4.1 Upravljački orijentisani modeli motora

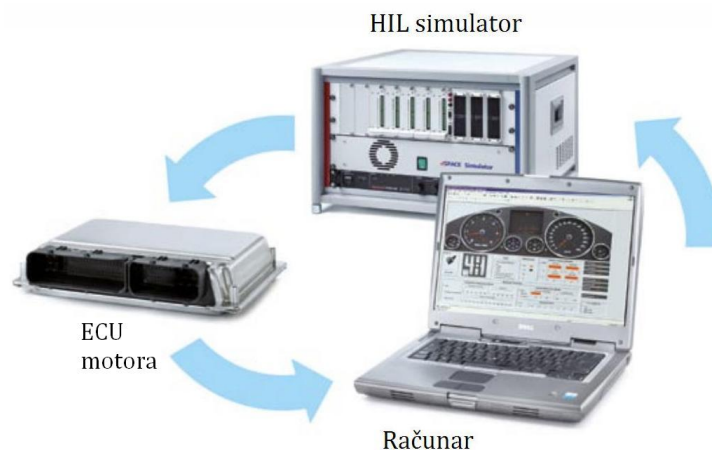
Upravljački orijentisani modeli nose takav naziv, jer se koriste u razvoju sistema upravljanja modernim motorima. Najčešće se ovi modeli koriste u fazi testiranja elektronskih upravljačkih jedinica (ECU) u tzv. *Hardware-in-the-Loop* (HIL) simulatorima [73]. Najvažnije komponente koje se međusobno sprežu prilikom testiranja elektronske upravljačke jedinice motora pomoću HIL simulacije, prikazane su na Sl. 4.1.

Kako se vidi na slici, centralno mesto u instalaciji za testiranje ECU zauzima HIL simulator. To je jedna hardverska platforma koja sadrži kontroler, koji radi u realnom vremenu<sup>18</sup>, kao i odgovarajuće ulazno-izlazne interfejsne. Ovi interfejsi omogućavaju povezivanje HIL simulatora sa ECU koja se želi testirati. U HIL simulatoru je softverski implementiran upravljački orijentisani model motora, koji tokom izvođenja simulacije oponaša rad motora u realnom vremenu. Na osnovu jednačina modela motora, kontroler HIL simulatora izračunava odgovarajuće veličine u realnom vremenu (na primer: broj obrtaja motora, maseni protok vazduha i sl.), pri čemu se one odmah transformišu u odgovarajuće analogne ili digitalne signale, pomoću generatora signala, koji su takođe deo

---

<sup>18</sup> eng. *Real-time controller*

HIL simulatora. Na taj način, oponašaju se signali senzora koji postoje na stvarnom motoru, tako da ECU „misli“ da je povezana sa motorom. Na osnovu primljenih signala sa „senzora“ iz HIL simulatora, ECU motora vrši određena izračunavanja prema svojim upravljačkim algoritmima i generiše odgovarajuće komandne signale. Ovi signali bi u realnom sistemu (u vozilu) trebalo da se šalju aktuatorima (kao što su, na primer, brizgači goriva, EGR ventil i sl.) koji treba da izvrše zadate komande. Međutim, pri testiranju ECU, komandni signali se šalju ponovo u HIL simulator, gde se transformišu u numeričke vrednosti odgovarajućih fizičkih veličina, koje može da prihvati matematički model motora u HIL simulatoru. Kao što se vidi na *Sl. 4.1*, i ECU i HIL simulator povezani su sa personalnim računarom (laptop ili desktop). Na ovom računaru instalirani su softverski alati koji omogućavaju konfigurisanje i podešavanje kompletne instalacije za testiranje ECU, parametrizaciju modela motora, vizualizaciju rezultata testiranja itd.



*Sl. 4.1 Testiranje ECU motora pomoću Hardware-in-the-Loop simulacije [74]*

Korišćenje HIL simulacije u testiranju ECU ima mnoge pozitivne strane, od kojih su najvažnije [75]:

- omogućava razvoj sistema upravljanja motorom paralelno sa razvojem ostalih podsistema motora, čime se skraćuje ukupno vreme potrebno za razvoj motora kao celine,
- omogućava testiranje ECU u ekstremnim uslovima, koji se ne mogu tako lako obezbediti, ako bi ECU prilikom testiranja bila povezana sa stvarnim motorom,
- omogućava simulacije otkaza i grešaka u motoru, bez rizika od štetnih posledica po sâm motor i
- omogućava precizno ponavljanje simulacija koliko god puta je to potrebno.

S obzirom na pobražano, kao i na činjenicu da svakim modernim motorom u vozilu upravlja elektronska upravljačka jedinica, primena HIL simulacije u testiranju ECU motora je veoma masovna. Zbog toga i upravljački orijentisani modeli motora, na osnovu kojih se ta simulacija izvodi, imaju veliki značaj.

S obzirom na specifično ciklično odvijanje radnih procesa u klipnim motorima, upravljački orijentisani modeli ovih motora razvrstavaju se na sledeće dve grupe [72]:

- **osrednjeni modeli** (eng. Mean Value Models - MVM) i
- **modeli zasnovani na diskretnim događajima** (eng. Discrete Event Models - DEM) ili trenutni modeli.

U poslednje vreme pojavljuju se i upravljački orijentisani modeli koji predstavljaju kombinaciju ove dve vrste i koriste dobre strane svake od njih, kako se može videti u referencama [76], [77], [78] i [79].

Osrednjeni modeli se primenjuju pri modelovanju onih procesa motora koji se ili potpuno kontinualno odvijaju ili taj kontinuitet nije značajnije narušen diskretnim



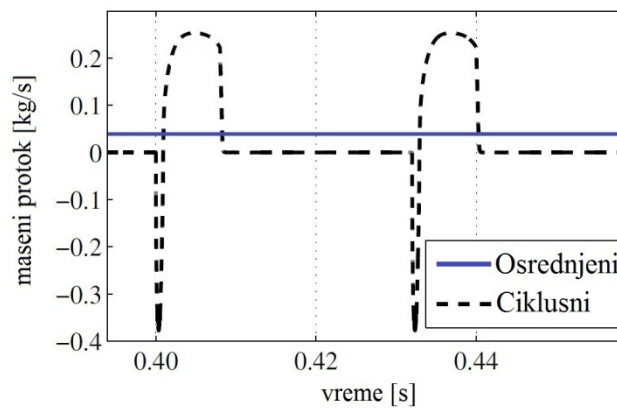
događajima, pa se ti događaji mogu zanemariti pri modelovanju. Primeri ovakvog modelovanja su modelovanje dinamike punjenja usisnog kolektora, ubrzavanja kolenastog vratila, dinamike turbokompresora itd., a mogu se naći u referencama [80], [81], [82], [83] i [84].

Svi podsistemi koji se modeluju pomoću osrednjenih modela motora, mogu se modelovati i pomoću trenutnih modela, ukoliko se želi pratiti kako diskretni događaji iz nekih drugih podсистema utiču na kontinualne procese u modelovanim podсистemima. Međutim, u tom slučaju modeli postaju znatno složeniji, tako da se trenutni modeli primenjuju uglavnom u modelovanju podсистema i procesa u kojima su dominantni diskretni događaji. Takav je slučaj, na primer, sa modelovanjem ubrizgavanja goriva, paljenja smeše, sagorevanja, izmene radne materije u pojedinim cilindrima itd., [85] i [86]. To su sve karakteristični događaji koji se pojavljuju u svakom radnom ciklusu motora, u svakom cilindru, pa se modeli zasnovani na diskretnim događajima u literaturi ponekad nazivaju i **ciklusnim modelima** [87], što će i ovde u daljem tekstu biti slučaj. Dakle, u ciklusnim modelima uzima se u obzir periodični karakter odvijanja radnih procesa u radnim ciklusima. Vremenski opseg u kojem se dešavaju promene praćenih veličina nije duži od vremena trajanja jednog radnog ciklusa. Zato je osnovna nezavisno promenljiva veličina u jednačinama ciklusnih modela ugao obrtanja kolenastog vratila. Osim toga, svi karakteristični događaji moraju biti sinhronizovani u odnosu na ugao obrtanja kolenastog vratila. Međutim, pošto se ovaj rad u svom najvećem delu bavi dešavanjima u usisnom sistemu motora, gde se sve promene odvijaju znatno sporije nego što su one u cilindrima, u daljem tekstu se fokus prebacuje na osrednjene modele motora, koji su u tom slučaju pogodniji za primenu, nego ciklusni modeli.

## 4.2 Osrednjeni modeli motora

Osrednjeni modeli motora imaju za cilj da predvide srednje vrednosti nekih važnih veličina koje opisuju stanje motora i koje se menjaju u vremenu. Jedan od osnovnih preduslova za primenu osrednjenih modela motora je taj, da je vremenski opseg, u kojem se dešavaju promene posmatranih veličina, znatno duži od vremena potrebnog za odvijanje jednog radnog ciklusa u cilindru. Zbog toga je u jednačinama osrednjenih modela motora nezavisno promenljiva veličina vreme, a ne ugao obrtanja kolenastog vratila. U osrednjenim modelima motora se zanemaruje pulsirajući karakter funkcionisanja motora, tj. ne uzima se u obzir periodično odvijanje radnih ciklusa u pojedinim cilindrima prema redosledu paljenja. Dakle, osrednjeni modeli motora su pojednostavljeni dinamički modeli koji omogućavaju računanje srednjih, a ne trenutnih vrednosti promenljivih u motoru, za razliku od ciklusnih modela.

Da bi stvari bile malo jasnije, ovde će biti dat primer usisnog sistema motora, gde su najvažnije promenljive koje se prate: maseni protok vazduha, apsolutni pritisak u usisnom kolektoru i temperatura u usisnom kolektoru. Sve one menjaju se periodično tokom odvijanja radnih ciklusa motora kroz različite dinamičke režime zbog otvaranja i zatvaranja ventila motora. Osrednjeni model motora ne uzima u obzir ove različite režime u svakom ciklusu ili pulsiranje trenutnog protoka, nego opisuje postepene promene srednjih vrednosti promenljivih tokom odvijanja radnih ciklusa uzimanjem u obzir srednjeg protoka. Ovo je ilustrovano na *Sl. 4.2*, gde se vide varijacije protoka vazduha pored leptira koje daje ciklusni model tokom dva radna ciklusa motora i odgovarajuća srednja vrednost protoka koju daje osrednjeni model motora.



Sl. 4.2 Maseni protok vazduha pored leptira po osrednjenom i ciklusnom modelu motora [30]

Slična situacija bi bila i ukoliko bi se pomoću osrednjenog i ciklusnog modela prikazala promena neke druge veličine npr. obrtnog momenta.

Postoji još nešto što je karakteristično za osrednjeni model motora, a što bi ovde trebalo pomenuti. Sa aspekta matematičke određenosti veza između njegovih ulaznih i izlaznih veličina, osrednjeni model motora predstavlja kompromis između modela tipa „crne kutije“<sup>19</sup> i modela tipa „bele kutije“<sup>20</sup>. Modeli tipa „crne kutije“ primenjuju se u slučajevima kada nema informacija o modelovanom sistemu odnosno nije poznata analitička zavisnost između izlaza i ulaza, niti se ona može izvesti na osnovu poznatih fizičkih zakona. Modeli tipa „bele kutije“ primenjuju se kada su dostupne sve potrebne informacije o modelovanom sistemu. Drugim rečima, u modelima tipa „bele kutije“ poznata je fizička zakonitost koja povezuje ulazne i izlazne veličine i ona je u potpunosti matematički opisana. Osrednjeni modeli motora mogu se svrstati u modele tipa „sive kutije“<sup>21</sup> [72] jer sadrže, kako egzaktno matematičke jednačine (što odgovara principu „bele kutije“), tako i empirijski ustanovljene veze između pojedinih veličina (ulazno-izlazni princip „crne kutije“). Ovo znači da je za formiranje jednog osrednjenog modela motora neophodno ispitivanje stvarnog motora na probnom stolu. Pri tome se snimaju neophodni podaci na relativno velikom broju ustaljenih režima rada motora, kako bi se dobio što veći broj tačaka za formiranje empirijske funkcionalne zavisnosti između pojedinih veličina. Postoji mogućnost da se osrednjeni model motora razvije i bez direktnog ispitivanja motora u laboratorijskim uslovima. To je moguće, ukoliko se sa visokom tačnošću simulira rad motora pomoću nekog drugog, detaljnijeg modela [88]. Ova druga opcija se takođe često primenjuje u istraživanjima u oblasti upravljački orijentisanih modela motora, naročito u poslednje vreme, kada postoje profesionalni softverski alati koji se koriste u automobilske industriji i pomoću kojih se uzima u obzir veliki broj uticajnih faktora, [89] i [90].

Treba razjasniti još neke pojmove, pre nego što bude prikazan jedan konkretan osrednjeni model motora. **Ulazne promenljive** modela motora su one koje se mogu podešavati da bi se upravljalo motorom. Takve su, na primer ugao leptira (otvor leptira), maseni protok goriva koje se ubrizgava, ugao pretpaljenja. **Promenljive stanja** su promenljive koje se dobijaju rešavanjem diferencijalnih jednačina korišćenih za opisivanje motora. Izbor ovih promenljivih u velikoj meri utiče na to, koliko će komplikovane biti diferencijalne jednačine i koliko će pomoćnih algebarskih jednačina biti potrebno da bi se dobio jedan koherentan dinamički model motora. Primeri ovakvih promenljivih mogu da budu apsolutni pritisak u usisnom kolektoru, temperatura u usisnom kolektoru, brzina obrtanja kolenastog vratila. Ukoliko su promenljive stanja motora merljive, one predstavljaju **izlazne promenljive** modela motora. **Unutrašnje promenljive** motora, kao

<sup>19</sup> eng. *Black-box model*

<sup>20</sup> eng. *White-box model*

<sup>21</sup> eng. *Gray-box model*

što su indikatorski stepen iskorišćenja, stepen punjenja i sl. određuju se kao funkcije promenljivih stanja. U osrednjenim modelima motora postoje dve vrste veza između promenljivih: **trenutne veze** i veze koje se postepeno uspostavljaju (za ove druge će, zbog jednostavnosti, biti uveden pojam **vremenske veze**). Razlika između ove dve vrste veza potiče od vremena koje je potrebno da bi se uspostavila ustaljena vrednost jedne promenljive, nakon izmene vrednosti neke druge promenljive veličine. U tom smislu, trenutne veze su one, kod kojih se ravnotežno stanje uspostavlja već posle nekoliko radnih ciklusa motora i one se izražavaju preko algebarskih jednačina. Primer trenutne veze je veza između masenog protoka vazduha pored leptira i otvora leptira. Za uspostavljanje ravnotežnog stanja u vremenskim vezama potrebno je od nekoliko desetina pa i do 1000 radnih ciklusa motora [80]. Ovakve veze se u osrednjenom modelu motora opisuju diferencijalnim jednačinama, a kao primer može se navesti veza između broja obrtaja motora i otvora leptira.

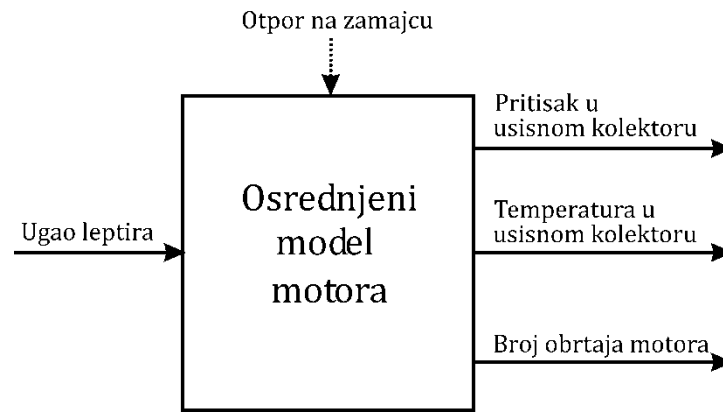
Sve u svemu, osrednjeni modeli motora su jednostavniji od ciklusnih modela motora, manje su zahtevni sa aspekta utrošenog računarskog vremena, pa su i pogodniji za implementaciju u kontrolerima koji rade u realnom vremenu. Bez obzira na njihovu jednostavnost, osrednjeni modeli motora postižu zadovoljavajuću tačnost u onim oblastima namene za koje su predviđeni. Kako je u prethodnom delu teksta rečeno, najvažnija oblast primene ovih modela motora je u razvoju sistema upravljanja motorom. Međutim, s obzirom na to, da osrednjeni modeli motora mogu da rade u realnom vremenu, pomoću njih je moguće simulirati greške u radu motora, pa sve veću primenu nalaze i u oblasti dijagnostikovanja motora u realnom vremenu. Pošto je u ovom radu reč o dijagnostikovanju neispravnosti u usisnom sistemu motora, ideja je da se jedan konkretan osrednjeni model motora iskoristi kao polazna osnova za razvoj sistema za automatsko simuliranje nekih od tih neispravnosti. Ovde je izbor pao na Hendriksov osrednjeni model automobilskog oto motora, iz dva razloga:

- Tačnost rezultata koji se dobijaju pomoću ovog modela na ustaljenim i na neustaljenim režimima rada motora iznosi  $\pm 2\%$ , što je na nivou greške merenja u laboratorijskim uslovima;
- Autor je obezbedio slobodan pristup tom svom modelu implementiranom u Matlab/Simulink formatu.

U nastavku će biti dat sažeti prikaz ovog modela motora, a detaljna izvođenja, u kojima se polazi od zakona o održanju mase i zakona o održanju energije, dostupna su u referencama [80], [91], [92] i [93]. Treba napomenuti da je ovaj Hendriksov model nastao pre pojave masovnije primene oto motora sa direktnim ubrizgavanjem, tako da je testiran na motorima sa centralnim ubrizgavanjem (SPI – Single Point Injection) i ubrizgavanjem u usisne kanale (MPI – Multi Point Injection), kako istovremenim tako i sekvencijalnim, [91] i [94].

### 4.3 Hendriksov osrednjeni model motora

Hendriksov osrednjeni model automobilskog oto motora je dinamički model i ima samo jednu ulaznu veličinu – ugao leptira, a izlazne veličine su pritisak i temperatura u usisnom kolektoru i broj obrtaja motora, kako se vidi na *Sl. 4.3*. Dakle, ovaj model izračunava kako se menjaju srednje vrednosti pomenutih izlaznih veličina u vremenu, sa povećanjem ili smanjenjem otvora leptira. Osim otvora leptira, otpor transmisije automobila takođe značajno utiče na dinamiku izlaznih veličina, ali se ne može svrstati u ulazne veličine, jer spoljašnji otpor nije veličina koja se može podešavati kako bi se upravljalo motorom. Zbog toga se otpor transmisije modeluje kao poremećaj koji ometa rad motora u odnosu na njegovo rasterećeno stanje.

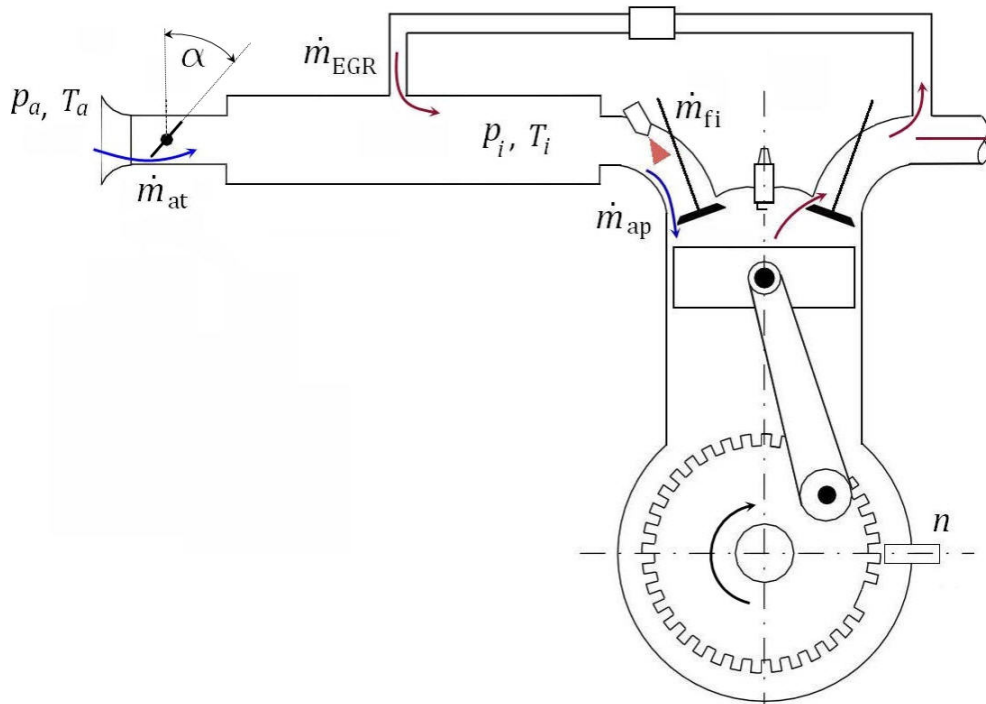


Sl. 4.3 Ulazne i izlazne veličine Hendriksovog modela

Ovaj model motora nastao je na osnovu razmatranja o protoku energije u motoru, gde se preko usisnog vazduha i goriva energija dovodi u motor, a iz njega se odvodi u vidu mehaničkog rada na zamajcu, kao i preko toplotnih i mehaničkih gubitaka. Zato je ovaj model sačinjen od tri dinamička podsistema, koji opisuju sledeće procese u motoru:

- proticanje goriva kroz usisni sistem (od brizgača do cilindra),
- proticanje vazduha (sa recirkulisanim izduvnim gasovima) kroz usisni sistem i
- ubrzavanje kolenastog vratila.

Uprošćeni šematski prikaz jednog oto motora sa multipoint ubrizgavanjem goriva dat je na Sl. 4.4, gde su prikazani oni podsistemi koji su obuhvaćeni Hendrikovim osrednjenim modelom motora. Na slici su još označene i neke važnije fizičke veličine relevantne za model, a malo kasnije, kada budu date odgovarajuće jednačine, biće i pojašnjene.



Sl. 4.4 Uprošćeni prikaz jednog oto motora sa istaknutim podsistemima Hendriksovog osrednjenog modela

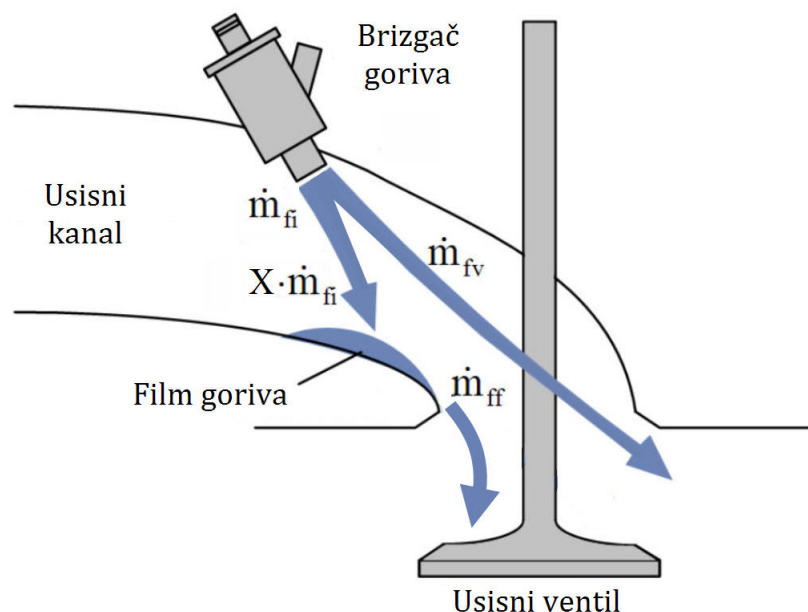
### 4.3.1 Podsystem proticanja goriva kroz usisni sistem

Ovaj podsystem je veoma bitan za ponašanje modela u prelaznim režimima rada motora, naročito pri bržim promenama opterećenja motora. Bitan je zbog pojave koja se zove kvašenje zidova<sup>22</sup> usisnog kolektora odnosno usisnih kanala. Ova pojava je karakteristična za motore sa centralnim ubrizgavanjem i motore sa ubrizgavanjem u usisne kanale i ima značajan uticaj na vrednost koeficijenta viška vazduha u cilindrima motora [95]. Naime, ako se gorivo ne ubrizgava direktno u cilindar, nego u struju vazduha koja protiče kroz usisni sistem, jedan deo tog goriva odmah isparava, a drugi deo dospeva na zidove usisnog kanala u vidu tankog filma goriva, kako je prikazano na Sl. 4.5. za jedan motor sa MPI sistemom napajanja. Zatim film goriva određenom brzinom isparava u nadolazeću struju vazduha i odlazi zajedno sa njom u cilindre, narušavajući na taj način predviđeni sastav smeše za aktuelni režim rada motora. Zbog toga se moraju uzeti u obzir i količina goriva koja formira film na zidovima usisnih kanala i količina goriva koja iz njega isparava, da bi se obezbedilo optimalno doziranje goriva. Ovo posebno važi za prelazne režime.

Tokom prelaznih režima koji su prouzrokovani povećanjem otvora leptira, mora se ubrizgati još dodatna količina goriva u odnosu na komandovanu količinu, da bi se uzela u obzir činjenica da značajan deo ubrizganog goriva ne dospeva odmah u cilindar. Ako ne bi bila ubrizgana dodatna količina goriva, kao rezultat bi nastala siromašna smeša, jer bi značajna količina goriva završavala u filmu na zidovima usisnog kanala. Naime, pri povećanju otvora leptira dolazi do trenutnog porasta pritiska u usisnom sistemu, pa je isparavanje goriva sa zidova usisnog sistema slabije. Smeša u cilindru bi bila siromašnija od potrebne, zbog veće količine tečnog goriva na zidovima usisnog sistema. Otuda je neophodna kompenzacija u vidu povećane količine ubrizganog goriva.

Obrnuto, tokom prelaznih režima koji su prouzrokovani smanjenjem otvora leptira, dolazi do umanjjenja pritiska u usisnom sistemu, što intenzivira isparavanje goriva sa zidova usisnog sistema. Zbog toga, u cilindre bi ulazila smeša koja je bogatija nego što je potrebno, pa treba umanjiti dozu ubrizganog goriva.

Dakle, zbog pojave kvašenja zidova usisnog sistema motora, dešava se da količina goriva, koja se nalazi u cilindru posle zatvaranja usisnih ventila, nije jednaka količini ubrizganog goriva, pa ovo treba da bude uzeto u obzir pri dinamičkom modelovanju motora.



Sl. 4.5 Šematski prikaz dinamike strujanja goriva kroz usisni sistem

<sup>22</sup> eng. *Wall wetting*

Ispareli deo goriva dospeva u cilindar u istom radnom ciklusu za koji je ubrizgana doza goriva i namenjena. Sa aspekta osrednjenog modela motora, smatra se da se ovo dešava trenutno, pa se maseni protok isparelog goriva  $\dot{m}_{fv}$  [kg/s] može izraziti algebarskom jednačinom [91]

$$\dot{m}_{fv} = (1 - X) \cdot \dot{m}_{fi} \quad (4.1)$$

gde je:

$X$  - udeo ubrizganog goriva koje se deponuje u usisnom sistemu u vidu filma i

$\dot{m}_{fi}$  - maseni protok ubrizganog goriva [kg/s]

Ovde treba napomenuti da tačke iznad oznaka za masu „m“ u jednačini (4.1) ne govore da je u pitanju izvod po vremenu, već da se radi o masenim protocima. Zbog toga je to algebarska jednačina.

Film goriva postepeno isparava zbog toplih zidova usisnog kanala, a tako nastala para goriva istom dinamikom biva zahvaćena strujom usisnog vazduha i završava u cilindru. Ovde je namerno upotrebljena reč „postepeno“, jer kod zagrejanog motora, količina goriva koja u toku jednog ubrizgavanja dospe na zidove usisnog kanala, ispari za 0,25 do 1 sekundi [92]. To odgovara vremenu za koje se izvrši nekoliko desetina radnih ciklusa motora, pa sa aspekta osrednjenog modela motora proces isparavanja filma goriva nije trenutnan, već vremenski. Zbog toga je za njegovo opisivanje neophodna diferencijalna jednačina, a ona ima oblik [91]

$$\dot{m}_{ff} = \frac{1}{\tau_{ff}} (X \cdot \dot{m}_{fi} - \dot{m}_{ff}) \quad (4.2)$$

gde su:

$\dot{m}_{fi}$  - maseni protok ubrizganog goriva [kg/s],

$\dot{m}_{ff}$  - maseni protok filma goriva [kg/s] i

$\tau_{ff}$  - vremenska konstanta isparavanja filma goriva (mada nema konstantnu vrednost) [s].

Konačno, maseni protok goriva pored usisnog ventila,  $\dot{m}_f$ , može se izraziti kao

$$\dot{m}_f = \dot{m}_{fv} + \dot{m}_{ff} . \quad (4.3)$$

Ovaj maseni protok predstavlja izlaznu veličinu podsistema proticanja goriva kroz usisni sistem, koji je tema ovog odeljka. Ulazna veličina je maseni protok ubrizganog goriva  $\dot{m}_{fi}$ , koji se određuje iz dobro poznatog izraza

$$\dot{m}_{fi} = \frac{\dot{m}_{ap}}{\lambda_{des} \cdot L_{th}} \quad (4.4)$$

gde je:

$\dot{m}_{ap}$  - maseni protok vazduha kroz usisne kanale,

$\lambda_{des}$  - željena vrednost koeficijenta viška vazduha i

$L_{th}$  - teorijski neophodna masa vazduha za potpuno sagorevanje jedinice mase goriva.

Jednačine (4.1), (4.2) i (4.3) čine opšti dinamički podmodel strujanja goriva u usisnom sistemu, koji važi i za singl-point i za multi-point sisteme ubrizgavanja goriva. Međutim, parametri u tim jednačinama,  $X$  i  $\tau_{ff}$ , zavise od režima rada motora, tj. od broja obrtaja motora  $n$  i od pritiska u usisnom kolektoru  $p_i$ , a ta zavisnost se određuje za svaki konkretan motor mapiranjem na osnovu eksperimentalnih ispitivanja pri prelaznim režimima rada. Radi ilustracije, za konkretan British Leyland četvorotaktni multi-point oto motor, radne zapremine 1275 cm<sup>3</sup>, empirijski utvrđeni izrazi za parametre  $X$  i  $\tau_{ff}$  su [96]

$$\tau_{ff}(p_i, n) = 1.35 \cdot (1.68 - 0.672n) \cdot (p_i - 0.825)^2 + (0.15 - 0.06n) + 0.56 \quad (4.5)$$

$$X_f(p_i, n) = -0.277p_i - 0.055n + 0.6 \quad (4.6)$$

gde veličine  $p_i$  i  $n$  treba da budu uvrštene u barima i hiljadama obrtaja u minutu, respektivno.

Dakle, da bi u svakom trenutku bile poznate vrednosti parametara  $X$  i  $\tau_{ff}$ , neophodno je pratiti promenljive stanja motora,  $p_i$  i  $n$ , u realnom vremenu. Da bi to bilo moguće, potrebne su dodatne diferencijalne jednačine, a one će biti date u sledećim odeljcima, gde će biti opisana preostala dva podsistema osrednjenog modela motora. Takođe, tamo će biti prikazano i kako se određuje maseni protok vazduha kroz usisne kanale  $\dot{m}_{ap}$ , neophodan za određivanje potrebnog masenog protoka ubrizganog goriva.

### 4.3.2 Podsystem dinamike strujanja kroz usisni kolektor

Dinamika strujanja kroz usisni kolektor zasniva se na principu punjenja i pražnjenja, kako je prikazano na *Sl. 4.4*. Slika prikazuje da u punjenju usisnog kolektora učestvuju maseni protok vazduha oko leptira  $\dot{m}_{at}$  i maseni protok recirkuliranih izduvni gasova  $\dot{m}_{EGR}$ , a u njegovom pražnjenju, maseni protok kroz usisne kanale  $\dot{m}_{ap}$ . Kada je u pitanju usisni sistem motora, osnovne promenljive stanja su pritisak i temperatura u usisnom kolektoru,  $p_i$  i  $T_i$ . Pri promenama ovih veličina, ravnotežno stanje se ne uspostavlja trenutno, već posle nekoliko desetina radnih ciklusa motora, pa su za njihovo određivanje neophodne diferencijalne jednačine [93], koje će ovde biti nazvane jednačinama stanja u usisnom kolektoru. Ove jednačine su izvedene na pretpostavci da su procesi punjenja i pražnjenja usisnog kolektora adijabatski, kao i da kroz usisni kolektor struji idealni gas.

Jednačina stanja pritiska u usisnom kolektoru izvedena je diferenciranjem pritiska u jednačini stanja idealnog gasa po vremenu, uz primenu zakona o održanju mase u usisnom kolektoru, tako da je njen konačan oblik [92]

$$\dot{p}_i = \frac{\kappa \cdot R}{V_i} \cdot (\dot{m}_{at} \cdot T_a + \dot{m}_{EGR} \cdot T_{EGR} - \dot{m}_{ap} \cdot T_i) \quad (4.7)$$

gde je

$\kappa$  - eksponent adijabate za vazduh,

$R$  - gasna konstanta,

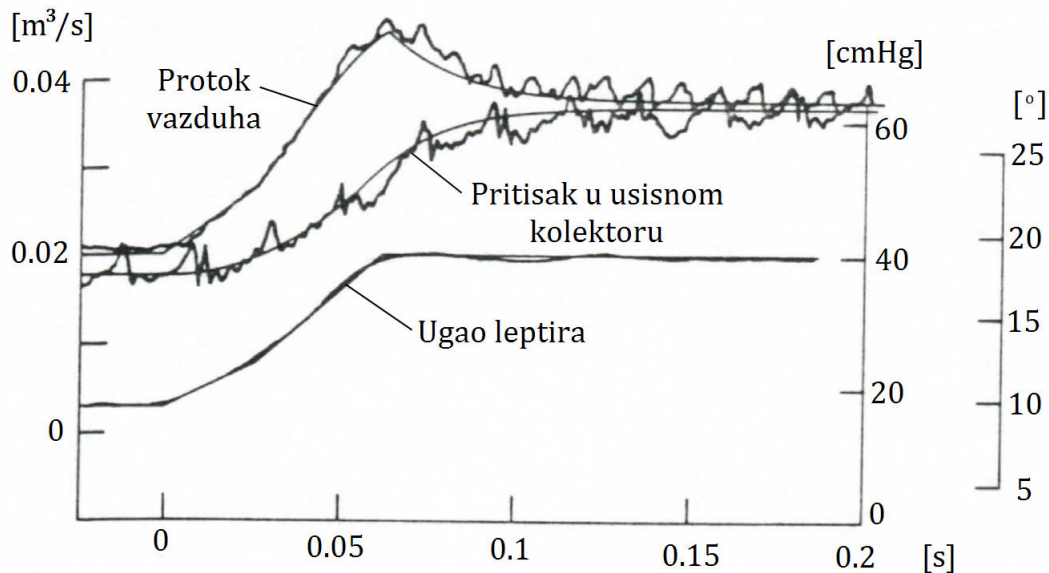
$V_i$  - zapremina usisnog kolektora,

$T_a$  - temperatura okoline i

$T_{EGR}$  - temperatura recirkuliranih izduvni gasova.

Za razliku od pritiska u usisnom kolektoru, protok vazduha oko leptira menja se gotovo trenutno sa promenom položaja leptira. Ovo je ilustrovano na *Sl. 4.6*, gde se vidi da se promena protoka vazduha odvija praktično istovremeno sa promenom ugla leptira, dok promena pritiska u usisnom kolektoru kasni. Zbog toga se maseni protok pored leptira u osrednjenom modelu motora svrstava u trenutne promenljive motora, te za njeno određivanje nije potrebna diferencijalna jednačina.

Međutim, bez obzira što će se za određivanje masenog protoka oko leptira koristiti algebarska jednačina, problem i dalje nije jednostavno rešiti, jer se protočni presek oko leptira menja u toku rada motora na neustaljenim režimima. Osim toga, koeficijent protoka je takođe promenljiva kategorija, jer se menja sa uglom leptira, odnosom pritisaka ispred i iza leptira, a zavisi i od Rejnoldsovog broja za leptir [95]. Dalje, brzina strujanja pored leptira na nekim režimima rada motora dostiže brzinu zvuka u vazduhu, tako da je ovde reč o strujanju kompresibilnog fluida.



Sl. 4.6 Promene protoka vazduha oko leptira i pritiska u usisnom kolektoru sa povećanjem otvora leptira za jedan V8 motor, radne zapremine  $5 \text{ dm}^3$  [95]

Generalno, strujanje vazduha pored leptira u većini slučajeva aproksimira se kao izentropsko strujanje kompresibilnog fluida kroz konvergentni ili konvergentno-divergentni mlaznik, kao u referencama [95] i [80], ali pokazalo se da ovakvi modeli imaju slabosti, kako u oblastima manjih snaga motora, tako i u blizini punog otvora leptira. Zbog toga je originalna jednačina za određivanje masenog protoka pored leptira, koja je izvedena u [80] i zasnivala se na pretpostavci o strujanju kompresibilnog fluida kroz konvergentni mlaznik, pretrpela određene izmene. U suštini, ovde je, između ostalog, došao do izražaja princip „sive kutije“, pominjan u odeljku 4.2, pa su autori Hendriksovog osrednjenog modela motora iskoristili, kako neke fizičke zakonitosti vezane za strujanje kompresibilnog fluida, tako i eksperimentalna merenja, da bi došli do jedne regresione jednačine za određivanje masenog protoka oko leptira. Konačan oblik te jednačine dat je u [92] i glasi

$$\dot{m}_{at}(\alpha, p_r) = \dot{m}_{at1} \cdot \beta(\alpha, p_r) + \dot{m}_{at0} \quad (4.8)$$

gde je:

$\alpha$  - ugao leptira u odnosu na ravan normalnu na pravac strujanja (vidi Sl. 4.4), u daljem tekstu: ugao leptira,

$p_r$  - odnos pritiska u usisnom kolektoru i pritiska okoline,  $p_r = \frac{p_i}{p_a}$ ,

$\beta(\alpha, p_r)$  - funkcija koja će malo kasnije biti objašnjena i

$\dot{m}_{at1}$  i  $\dot{m}_{at0}$  su konstante dobijene na osnovu eksperimentalnih rezultata.

U eksperimentalnim ispitivanjima mereni su maseni protok vazduha oko leptira i osnovne veličine koje imaju uticaja na taj protok: ugao leptira, pritisci ispred i iza leptira i temperatura okoline, u velikom broju stacionarnih radnih tačaka, a onda su „fitovanjem“ određene konstante  $\dot{m}_{at1}$  i  $\dot{m}_{at0}$ .

Iz jednačine (4.8) vidi se da postoji vrlo jednostavna zavisnost masenog protoka oko leptira i funkcije  $\beta(\alpha, p_r)$  - ona je linearna. Funkcija  $\beta(\alpha, p_r)$  ima sledeći oblik

$$\beta(\alpha, p_r) = \frac{p_a}{\sqrt{T_a}} \cdot \beta_1(\alpha) \cdot \beta_2(p_r) \quad (4.9)$$

gde je:

$$\beta_1(\alpha) = 1 - \cos(\alpha) - \frac{\alpha_0^2}{2} \quad (4.10)$$



$$\beta_2(p_r) = \begin{cases} \sqrt{1 - \left(\frac{p_r - p_c}{1 - p_c}\right)^2} & \text{ako je } p_r \geq p_c \\ 1 & \text{ako je } p_r < p_c \end{cases} \quad (4.11)$$

$\alpha_0$  – ugao zatvorenog leptira i

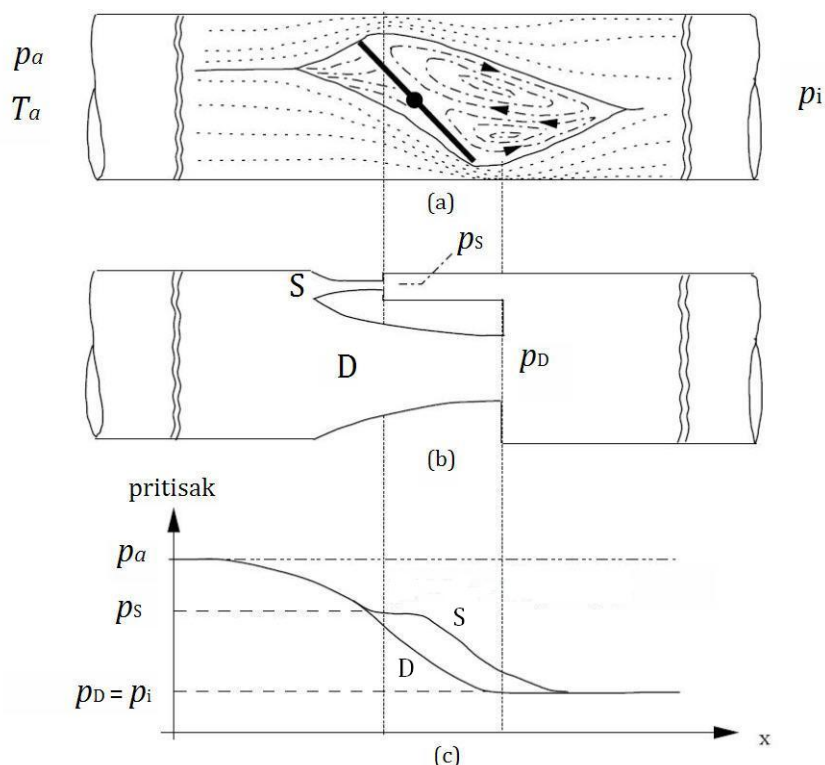
$p_c$  – kritični odnos pritisaka iza i ispred leptira.

Funkcija  $\beta_1(\alpha)$  opisuje uticaj promenljivog protočnog preseka oko leptira na maseni protok vazduha. Funkcija  $\beta_2(p_r)$  uzima u obzir činjenicu da brzina strujanja vazduha oko leptira ne može biti veća od brzine zvuka. Drugim rečima, postoji neka kritična vrednost odnosa pritiska iza i pritiska ispred leptira, pri kojem brzina strujanja dostiže brzinu zvuka i, ma koliko se odnos ta dva pritiska smanjivao ispod kritične vrednosti, brzina strujanja se više ne može povećavati. Ta kritična vrednost određena je izrazom [97]

$$p_c = \left(\frac{2}{\kappa+1}\right)^{\frac{\kappa}{\kappa-1}} \quad (4.12)$$

i za vazduh ima vrednost 0.528.

Još jedna specifičnost Hendriksovog modelovanja masenog protoka oko leptira je to, što je jednačina (4.9) izvedena na osnovu dva fizički odvojena paralelna izentropska toka. Ovo izvođenje je predstavljeno u [92] a u proširenom obliku i sa više detalja u [94], gde se može naći tačan izraz. Jednačina (4.11) je kompaktna aproksimacija tačnog izraza, koja je primenjena i u Hendriksovom generičkom osrednjenom modelu motora u matlab/simulink formatu. Ilustracija podele strujnog toka vazduha oko leptira na dva razdvojena protoka data je na Sl. 4.7a.



Sl. 4.7 Model strujanja oko leptira pomoću dva odvojena toka vazduha [94]

Veći protok je obeležen oznakom D i nazvan je dominantnim, a manji protok je obeležen oznakom S i nazvan je podređenim<sup>23</sup>, što je prikazano u vidu dva odgovarajuća konvergentna mlaznika na Sl. 4.7b. Promena pritiska u dominantnom i podređenom toku

<sup>23</sup> eng. *subordinate*

prikazana je na *Sl. 4.7c*. Efekat takve promene pritiska je proširenje opsega pritiska u usisnom kolektoru, pri kojima se strujanje oko leptira odvija brzinom manjom od brzine zvuka. Na ovaj način su poboljšani rezultati u oblasti manjih opterećenja motora, što je bila jedna od slabosti prvobitno razvijenog modela.

Sledeća komponenta, neophodna za rešavanje diferencijalne jednačine stanja pritiska u usisnom kolektoru, je maseni protok vazduha kroz usisne kanale,  $\dot{m}_{ap}$ . On se određuje na osnovu izraza proisteklog iz definicije koeficijenta punjenja  $\eta_v$  za četvorotakti motor [95]:

$$\eta_v = \frac{\dot{m}_{ap} \cdot 120}{V_d \cdot \rho_i \cdot n} \quad (4.13)$$

gde je:

$V_d$  - radna zapremina motora,

$\rho_i$  - gustina vazduha u usisnom kanalu i

$n$  - broj obrtaja kolenastog vratila u minutu.

Iz jednačine (4.13) i jednačine stanja idealnog gasa proizilazi sledeći izraz za maseni protok vazduha kroz usisne kanale

$$\dot{m}_{ap} = \eta_v \cdot \frac{V_d \cdot p_i}{R \cdot T_i} \cdot \frac{n}{120} \quad (4.14)$$

koji ima prilično jednostavan oblik. Međutim, problem je što koeficijent punjenja zavisi od režima rada motora, prvenstveno od broja obrtaja i od pritiska u usisnom kolektoru, a ne postoji analitički izveden izraz koji povezuje ove tri veličine. Zbog toga su u mnogim radovima autori određivali koeficijent punjenja na osnovu empirijski izvedenih regresionih funkcija, [98], [99], [100] i [101]. Hendricks i Sorenson su u referenci [80] takođe postupili na taj način. Eksperimentalnim merenjima na nekoliko različitih motora, pokazali su da regresiona funkcija koeficijenta punjenja oblika

$$\eta_v(n, p_i) = a_{n0} + a_{n1} \cdot n + a_{n2} \cdot n^2 + a_{p1} \cdot p_i \quad (4.15)$$

veoma dobro aproksimira zavisnost koeficijenta punjenja od broja obrtaja motora i pritiska u usisnom kolektoru, u širokom opsegu režima rada motora. Međutim, za vrednosti  $p_i$  niže od 0.4 bara, greška postaje veća od 2% [92]. Osim toga, s obzirom na dve promenljive u jednačini (4.15), potrebno je izvršiti merenja u velikom broju radnih tačaka motora, da bi se odredili regresioni koeficijenti  $a_{n0}$ ,  $a_{n1}$ ,  $a_{n2}$  i  $a_{p1}$ . Sve ovo komplikuje određivanje masenog protoka vazduha kroz usisne kanale, naročito kada to treba uraditi za neki drugi motor ili motor sa promenjenom konfiguracijom usisnog sistema, pa su neophodni novi eksperimenti mapiranja za dobijanje novih regresionih funkcija za koeficijent punjenja. Zato je bilo poželjno naći neki drugi, jednostavniji izraz za određivanje koeficijenta punjenja.

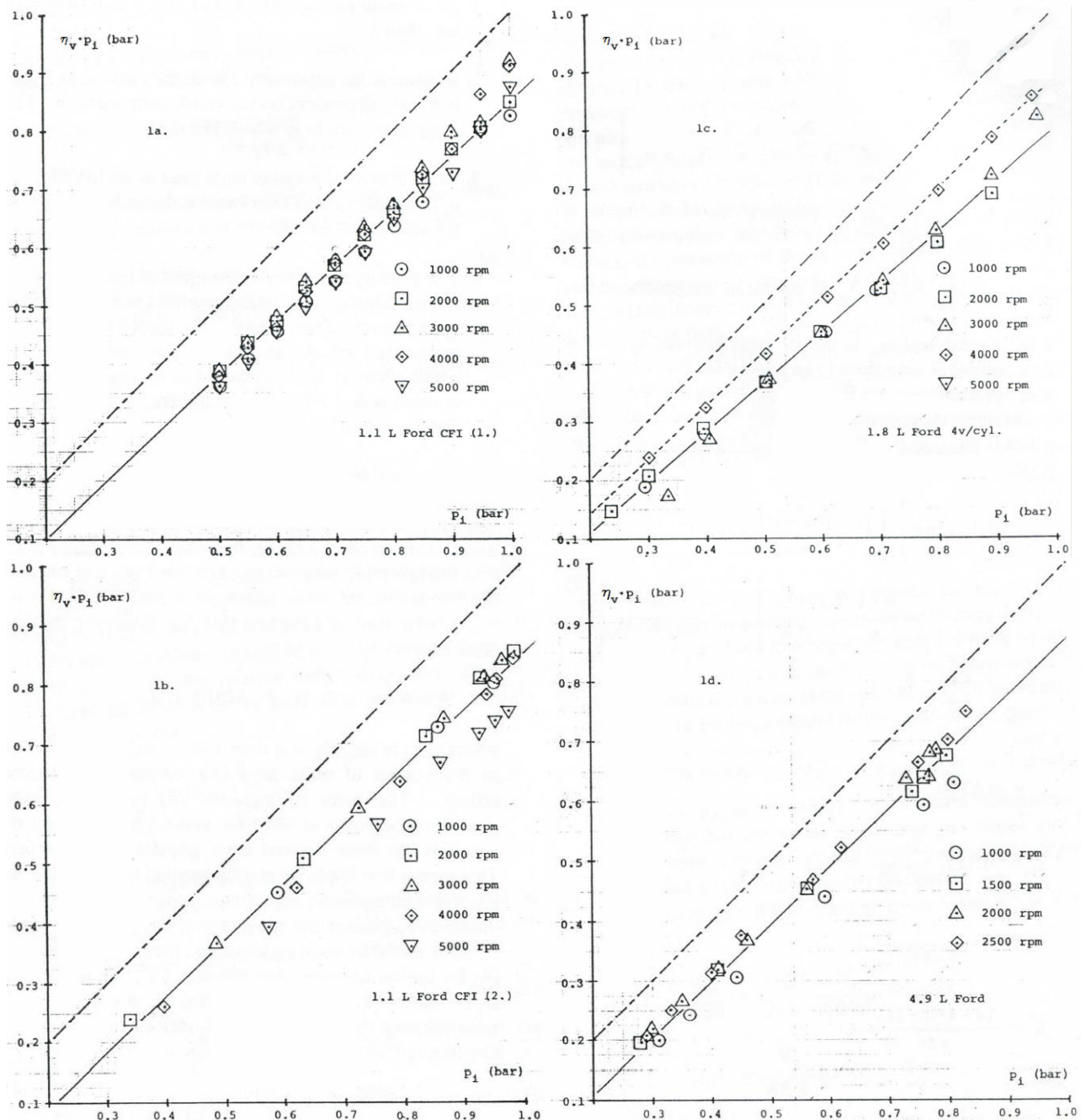
U tom cilju su Hendriks i saradnici. [92] napisali jednačinu (4.13) u obliku

$$\dot{m}_{ap} = \frac{V_d}{120 \cdot R \cdot T_i} \cdot (\eta_v \cdot p_i) \cdot n \quad (4.16)$$

i pokazali da je lakše modelovati izraz u zagradi  $(\eta_v \cdot p_i)$ , nego direktno koeficijent punjenja. Pri tome su primenili prvi zakon termodinamike na pumpni deo radnog ciklusa četvorotaktnog motora i došli do veoma jednostavnog konačnog izraza

$$(\eta_v \cdot p_i) = s_i(n) \cdot p_i - y_i(n) \quad (4.17)$$

gde su  $s_i$  i  $y_i$  slabe funkcije broja obrtaja i pri tome je  $y_i \ll s_i$ . Kompletno izvođenje jednačine (4.17) dato je u [92]. Tamo je pokazano i da su vrednosti  $s_i(n)$  vrlo malo manje od 1, a vrednosti  $y_i(n)$  iznose oko 0.1, bez obzira na broj obrtaja motora, a to je demonstrirano na primerima nekoliko veoma različitih konkretnih motora na *Sl. 4.8*.



Sl. 4.8 Primeri zavisnosti veličine  $(\eta_v \cdot p_i)$  od  $p_i$  za različite motore, pri različitim brojevima obrtaja [92]

Na slici se vidi da dva različita motora male radne zapremine sa centralnim ubrizgavanjem (1a i 1b), zatim motor srednje radne zapremine sa multipoint ubrizgavanjem i 4 ventila po cilindru (1c) i motor velike radne zapremine (1d) svi imaju približno isti koeficijent pravca ( $s_i$ ) i odsečak na ordinati ( $y_i$ ) u širokom dijapazonu brojeva obrtaja. Ovo omogućava da se maseni protok vazduha kroz usisne kanale odredi iz izraza

$$\dot{m}_{ap} = \frac{V_d}{120 \cdot R \cdot T_i} \cdot (s_i \cdot p_i - y_i) \cdot n \quad (4.18)$$

sa velikom tačnošću (posmatrano sa aspekta osrednjenog modela motora), bez potrebe da se ponovo izvode eksperimenti mapiranja pri modelovanju svakog novog motora.

Što se tiče masenog protoka recirkulisanih izduvni gasova  $\dot{m}_{EGR}$ , on zavisi od velikog broja fizičkih uticaja, pa je njegovo određivanje veoma komplikovano. U referenci [92] prikazano je određivanje ove veličine, ako EGR sistem koristi elektromagnetni EGR ventil

sa impulsnom regulacijom i dato je na nivou upravljanja ventilom preko dužine impulsa. Određivanje  $\dot{m}_{EGR}$  na tom nivou je bitno, samo ukoliko se osrednjeni model motora primenjuje u HIL simulacijama, što ovde nije slučaj. Kada je reč o primeni osrednjenog modela motora u simuliranju neispravnosti u usisnom sistemu pri ustaljenom radu motora, od interesa je malo globalniji pristup, koji nije vezan samo za jedan način upravljanja EGR sistemom. Otuda se, u takve svrhe, može posmatrati samo procentualni udeo masenog EGR protoka u masenom protoku vazduha kroz usisne kanale, koji maksimalno ide do oko 20%, pa će i u ovom radu biti pretpostavljena ta vrednost.

Pored do sada opisanih masenih protoka, za rešavanje jednačine (4.7) neophodno je i poznavanje temperature  $T_i$  u svakom trenutku vremena. Diferencijalna jednačina koja to omogućava, izvedena je u [92] primenom zakona o održanju mase, zakona o održanju energije i jednačine stanja idealnog gasa na otvoreni termodinamički sistem usisnog kolektora, i ima oblik

$$\dot{T}_i = \frac{R \cdot T_i}{p_i \cdot V_i} \cdot [\dot{m}_{at} \cdot (\kappa \cdot T_a - T_i) + \dot{m}_{EGR} \cdot (\kappa \cdot T_{EGR} - T_i) - \dot{m}_{ap} \cdot (\kappa - 1) \cdot T_i]. \quad (4.19)$$

Jednačine (4.7) i (4.19) su suštinski deo podsistema dinamike strujanja kroz usisni kolektor u Hendriksovom adijabatskom osrednjenom modelu motora.

### 4.3.3 Podsystem dinamike kolenastog vratila

Jednačina koja opisuje dinamiku kolenastog vratila u Hendriksovom osrednjenom modelu motora izvedena je diferenciranjem po vremenu zakona o održanju energije primenjenog na obrtno kretanje kolenastog vratila i ima oblik [80]

$$\dot{n} = \frac{1}{I \cdot n} \left[ \eta_i \cdot H_u \cdot \dot{m}_f(t - \tau_d) - (P_b(n) + P_f(n) + P_p(p_i, n)) \right] \quad (4.20)$$

gde je:

$I$  - skalirani moment inercije motora i radne mašine,

$\eta_i$  - indikatorski stepen iskorišćenja,

$H_u$  - donja toplotna moć goriva,

$t$  - vreme,

$\tau_d$  - srednje kašnjenje porasta obrtnog momenta u odnosu na trenutak ubrizgavanja,

$P_b$  - snaga potrebna za savladavanje otpora na zamajcu motora,

$P_p$  - snaga pumpnih gubitaka i

$P_f$  - snaga mehaničkih gubitaka.

Praktičnije je da jednačina (4.20) bude napisana preko broja obrtaja motora, a ne preko ugaone brzine, pa se u njoj pojavljuje skalirani moment inercije. Skalirani moment inercije je u vezi sa stvarnim momentom inercije motora i radne mašine  $I_{stv}$  preko izraza

$I = I_{stv} \cdot \left(\frac{2\pi}{60}\right)^2$ . Veličina  $\tau_d$  na neki način u osrednjeni model motora uvodi cikličnu prirodu odvijanja radnih procesa u motoru, jer uzima u obzir pojavu kašnjenja porasta obrtnog momenta u odnosu na trenutak poslednjeg ubrizgavanja goriva. Ovo kašnjenje se može aproksimirati kao srednje vreme između dva uzastopna upaljenja, što kod četvorotaktnog motora sa  $n_{cil}$  cilindara iznosi  $\tau_d = \frac{120}{n \cdot n_{cil}}$ .

Funkcije  $P_b(n)$ ,  $P_f(n)$ ,  $P_p(p_i, n)$  teško je analitički izvesti, jer ne postoji dovoljno razvijena teorija na osnovu koje bi se to moglo uraditi. Zbog toga se ove funkcije izražavaju preko empirijski ustanovljenih zavisnosti. Tako su još Taylor i Taylor [102] analiziranjem podataka dobijenih ispitivanjem velikog broja motora intuitivno došli do funkcija  $P_f(n)$  i  $P_p(p_i, n)$ , a kasnije je i statistički potvrđen, kao najbolji, sledeći oblik ove dve funkcije:

$$P_f = a_0 \cdot n + a_1 \cdot n^2 + a_2 \cdot n^3 \quad (4.21)$$

$$P_p = (a_3 \cdot n + a_4 \cdot n^2) \cdot p_i \quad (4.22)$$

gde su  $a_0$ ,  $a_1$ ,  $a_2$ ,  $a_3$  i  $a_4$  koeficijenti koji se dobijaju korišćenjem regresione analize podataka dobijenih eksperimentalnim putem. Oblik funkcije  $P_b(n)$  koji je korišćen u Hendriksovom osrednjenom modelu motora je

$$P_b = k_b \cdot n^3 \quad (4.23)$$

gde je  $k_b$  koeficijent spoljašnjeg opterećenja motora, koji se određuje na osnovu izmerenih vrednosti broja obrtaja i efektivne snage na većem broju ustaljenih režima rada motora.

Poslednja veličina koju treba odrediti da bi se mogla rešavati jednačina (4.20) je  $\eta_i$ , koja zavisi od velikog broja uticajnih faktora. Zbog toga je veoma teško na osnovu dosadašnjih teorijskih saznanja izvesti izraz koji povezuje indikatorski stepen iskorišćenja i te uticajne faktore, pa se mora pribeći fenomenološkom modelu. Drugim rečima, neophodno je korišćenje empirijskih saznanja o uticajima pojedinih veličina na  $\eta_i$ . Na sreću, moguće je izdvojiti četiri glavna, manje ili više nezavisna, uticaja: broj obrtaja motora  $n$ , pritisak u usisnom kolektoru  $p_i$ , koeficijent viška vazduha  $\lambda$  i ugao pretpaljenja  $\theta$ , a to se može napisati kao

$$\eta_i(n, p_i, \lambda, \theta) = \eta_i(n) \cdot \eta_i(p_i) \cdot \eta_i(\lambda, n) \cdot \eta_i(\theta, n, p_i). \quad (4.24)$$

Članovi sa desne strane znaka jednakosti u jednačini (4.24) navedeni su po redosledu važnosti, a njihovi empirijski ustanovljeni oblici su [92]

$$\eta_i(n) = a_0 \cdot (1 - a_1 \cdot n^{a_2}) \quad (4.25)$$

$$\eta_i(p_i) = b_0 + b_1 \cdot p_i + b_2 \cdot p_i^2 \quad (4.26)$$

$$\eta_i(\lambda, n) = c_0 + c_1 \cdot \lambda + c_2 \cdot \lambda^2 + c_3 \cdot n \quad (4.27)$$

$$\eta_i(\theta, n, p_i) = d_0 + d_1 \cdot (\theta - \theta_{\text{MBT}}) + d_2 \cdot (\theta - \theta_{\text{MBT}})^2 \quad (4.28)$$

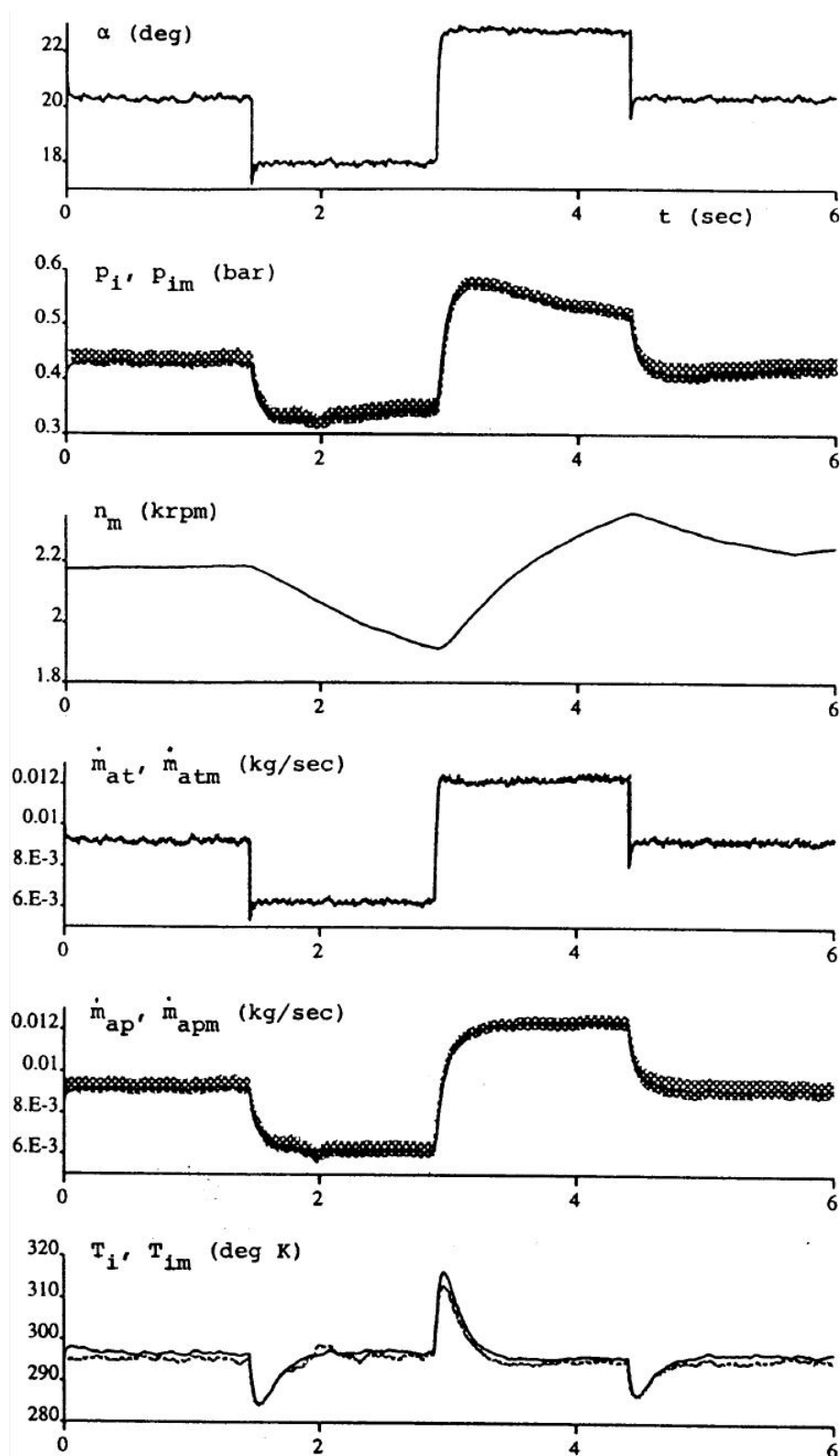
gde su  $a_0$ - $a_2$ ,  $b_0$ - $b_2$ ,  $c_0$ - $c_3$  i  $d_0$ - $d_2$  regresioni koeficijenti, a  $\theta_{\text{MBT}}$  je ugao pretpaljenja pri kojem se postiže maksimalni obrtni moment i koji zavisi od  $n$  i  $p_i$ .

Važno je istaći da su uticaji broja obrtaja u funkciji  $\eta_i(\lambda, n)$ , kao i broja obrtaja i pritiska u usisnom kolektoru u funkciji  $\eta_i(\theta, n, p_i)$  veoma slabi, pa se može smatrati da svaka od funkcija iz jednačina (4.25)-(4.28) zavisi samo od po jedne promenljive. To u mnogome olakšava određivanje regresionih koeficijenata, jer se znatno smanjuje broj eksperimentalnih radnih tačaka motora neophodnih za mapiranje.

Treba napomenuti da funkcija  $\eta_i(\theta, n, p_i)$  nije implementirana u generičkom Hendriksovom osrednjenom modelu motora koji je autor modela stavio na raspolaganje u Matlab/Simulink formatu, odnosno tamo je njena vrednost jednaka jedinici. Na taj način nije direktno uzet u obzir uticaj ugla pretpaljenja na indikatorski stepen iskorišćenja.

Sve u svemu, bez obzira na neka uprošćenja koja su učinjena u Hendriksovom osrednjenom modelu motora, njegovo slaganje sa izmerenim vrednostima relevantnih promenljivih veličina u konkretnom motoru koji je modelovan, je izuzetno dobro. Utvrđeno je odstupanje simuliranih u odnosu na izmerene vrednosti do oko  $\pm 2\%$  na ustaljenim režimima rada u celokupnom radnom opsegu modelovanog motora. Ova konstatacija može se proširiti i na neustaljene režime rada motora, što je potvrđeno nizom eksperimenata u različitim oblastima radnog opsega motora. Na Sl. 4.9 dati su dijagrami promene ugla leptira  $\alpha$ , pritiska u usisnom kolektoru  $p_i$ , broja obrtaja motora  $n$ , masenog protoka oko leptira  $\dot{m}_{\text{at}}$ , masenog protoka kroz usisne kanale  $\dot{m}_{\text{ap}}$  i temperature u usisnom kolektoru  $T_i$  tokom 6 sekundi, pri naglim promenama opterećenja motora. Slovo „m“ u indeksu

pobrojanih oznaka govori da se oznaka odnosi na izmerene vrednosti odgovarajuće veličine. Izmerene vrednosti su označene šrafiranim isprekidanim linijama, a vrednosti dobijene pomoću osrednjenog modela motora, označene su punim linijama.



Sl. 4.9 Izmerene vrednosti i vrednosti dobijene pomoću Hendriksovog osrednjenog modela, za neke bitnije veličine u usisnom sistemu, pri ustaljenom i neustaljenom radu motora [80]

Ovako dobri rezultati Hendriksovog osrednjenog modela motora daju za pravo da se on iskoristi u daljim istraživanjima kao da je u pitanju realan motor. Ideja je da se ovaj model

nadogradi uključivanjem dodatnih jednačina, pomoću kojih bi se mogle oponašati neke neispravnosti u usisnom sistemu motora, kao i da se te jednačine integrišu u Matlab/Simulink format već postojećeg generičkog Hendriksovog modela. Tako bi se dobio jedan simulator koji bi omogućio izvođenje velikog broja virtualnih eksperimenata u relativno kratkom vremenu, što bi dalje omogućilo da se razvije dijagnostički model za otkrivanje i prepoznavanje neispravnosti u usisnom sistemu. U sledećem poglavlju biće dat opis modela za oponašanje tih neispravnosti, zasnovan na Hendriksovom modelu prikazanom u odeljku 4.3.

# 5 SIMULATOR NEISPRAVNOSTI U USISNOM SISTEMU

## 5.1 Uvod

Ovde su od interesa neispravnosti koje se relativno teško otkrivaju, jer uglavnom ne dovode do prestanka rada motora, a većina njih ni do vidljive promene u radu motora. Imajući to u vidu, uzete su u obzir sledeće neispravnosti:

- pogrešno očitavanje senzora pritiska u usisnom kolektoru (MAP senzor <sup>24</sup>),
- pogrešno očitavanje senzora temperature u usisnom kolektoru (IAT senzor <sup>25</sup>),
- pogrešno očitavanje senzora protoka vazduha (MAF senzor <sup>26</sup>),
- postojanje nepredviđenog otvora u usisnom sistemu (ulaz tzv. „falš“ vazduha u usisni sistem) i
- umanjeni EGR protok.

S obzirom da je reč o usisnom sistemu, najvažniji parametar koji utiče na doziranje goriva jeste maseni protok vazduha. Taj protok se kod današnjih automobilskih motora određuje na dva načina: merenjem pomoću senzora protoka (ako motor poseduje taj senzor) ili računanjem prema tzv. „speed-density“ metodi (ukoliko motor nema protokomer vazduha). Simulator neispravnosti koji je ovde razvijen, može se primeniti bez obzira na način određivanja protoka vazduha kroz usisni sistem motora. Međutim, način određivanja protoka vazduha utiče na to, koje će neispravnosti biti posmatrane. Tako, na primer, kod motora koji nema protokomer vazduha, svakako neće biti interesantna neispravnost „pogrešno očitavanje MAF senzora“.

Kada je reč o pogrešnom očitavanju senzora, ne misli se na greške pri kojima signal senzora izlazi izvan svog mernog opsega ili iz realnog opsega u kojem se kreće merena veličina, jer se takve greške lako otkrivaju. Ovde se radi o relativno malim greškama očitavanja, koje je teško otkriti.

Pri razvoju simulatora neispravnosti, usvojene su sledeće pretpostavke:

- 1) motor je usisni, oto, sa indirektnim ubrizgavanjem goriva,
- 2) motor radi na ustaljenom režimu pri normalnoj radnoj temperaturi,
- 3) motor radi u režimu održavanja željene vrednosti koeficijenta viška vazduha,
- 4) broj neispravnosti u jednom trenutku ne može biti veći od jedan,
- 5) veličina potrebnog masenog EGR protoka na nekom režimu rada motora proporcionalna je veličini registrovanog masenog protoka kroz usisni kolektor na tom režimu.

Prve dve pretpostavke su ograničenja, u okviru kojih je simulator neispravnosti razvijan. Hendriksov osrednjeni model je razvijan za usisne motore sa centralnim ubrizgavanjem i multi-point ubrizgavanjem u usisne kanale, pa su i ovde, pri razvoju simulatora neispravnosti nasleđene te pretpostavke. Motori sa direktnim ubrizgavanjem imaju složeniji način doziranja goriva, koji nije opisan jednačinama Hendriksovog modela.

---

<sup>24</sup> eng. *Manifold Absolute Pressure sensor*

<sup>25</sup> eng. *Intake Air Temperature sensor*

<sup>26</sup> eng. *Mass Air Flow sensor*



Pošto je krajnji cilj ovog rada razvoj dijagnostičkog modela zasnovanog na podacima, a softverski alat koji je korišćen pri izradi modela, Matlab, ima određena ograničenja kada je u pitanju količina podataka kojima može da manipuliše, jasno je da bi uzimanje u obzir neustaljenih režima rada motora izazvalo velike probleme u istraživanju. Naime, postoji praktično beskonačan broj kombinacija početnog i krajnjeg otvora leptira kojima se menja režim rada motora. Još kad se doda i mogućnost da se te različite kombinacije mogu postići različitim brzinama otvaranja/zatvaranja leptira, zaključak je da je pretpostavka o radu motora na ustaljenom režimu neophodna. Na kraju krajeva, prelazni režim traje, dok se ne uspostavi neki novi ustaljeni režim, a tada može da se primeni dijagnostički model koji je pravljen za ustaljene režime. Istina, rađena su neka istraživanja na ovu temu, u kojima su razmatrani neustaljeni režimi [103], ali su ona bila ograničena na prilično mali broj prelaznih režima i mali broj neispravnosti.

Poslednje tri pretpostavke su uprošćenja koja se dobro uklapaju u stvarno funkcionisanje motora i obrazložene su u nastavku.

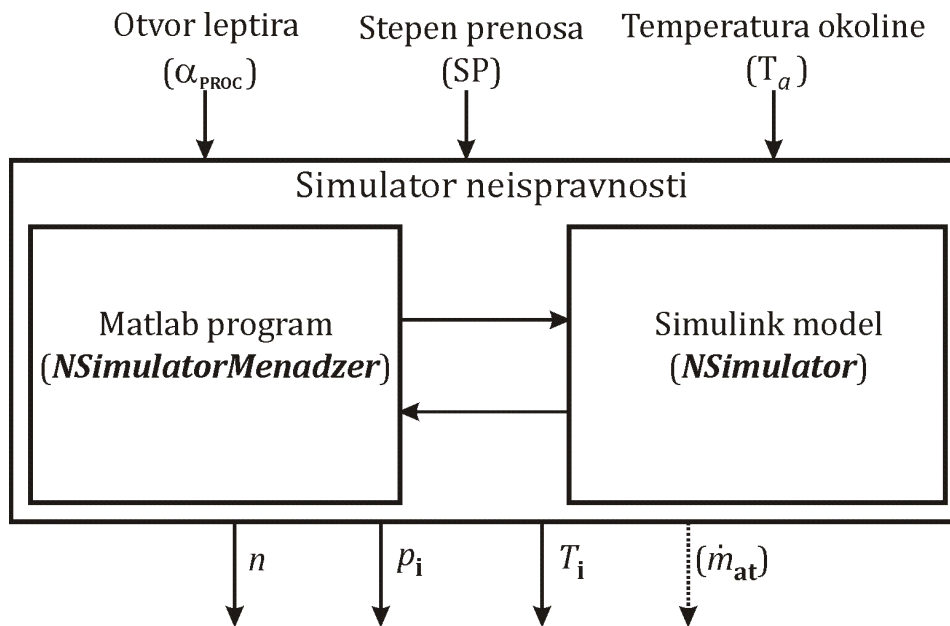
Željena vrednost koeficijenta viška vazduha najbolje se održava u zatvorenoj sprezi sa lambda-sondom i ta mogućnost se primenjuje kadgod su ispunjeni odgovarajući uslovi. Ako ti uslovi nisu ispunjeni, motor radi u tzv. otvorenoj sprezi sa lambda-sondom. Režim otvorene sprege sa lambda-sondom podrazumeva da se doziranje goriva sprovodi na osnovu željene vrednosti koeficijenta viška vazduha za trenutni režim rada motora, ali bez uzimanja u obzir signala pomenutog senzora. U najvećem delu svog perioda eksploatacije, automobilski motori rade u režimu održavanja željene vrednosti koeficijenta viška vazduha, bez obzira da li je u pitanju zatvorena ili otvorena sprega sa lambda-sondom. U ovoj činjenici treća od gore navedenih pretpostavki nalazi svoje opravdanje.

Pretpostavka o nemogućnosti postojanja dve neispravnosti istovremeno, prilično je realna, jer se smatra da će, kada se pojavi prva neispravnost, dijagnostički sistem relativno brzo da reaguje, pa se mogu preduzeti mere na njenom otklanjanju, pre nego što se eventualno pojavi i neka nova neispravnost. Što se tiče pretpostavki o funkcionisanju EGR sistema, situacija je slična. Poznato je da se najbolji rezultati u smanjenju sadržaja  $\text{NO}_x$  u izduvnim gasovima, na srednjim opterećenjima i srednjim brojevima obrtaja postižu kada se oko 15-25% izduvnih gasova usmeri u EGR vod, a to je otprilike i maksimalni udeo recirkuliranih izduvnih gasova sa kojim oto motor može bez problema da funkcioniše na tim režimima rada, [95] i [104]. Dakle, EGR protok je proporcionalan ukupnom protoku produkata sagorevanja kroz izduvni sistem. Sa druge strane, maseni protok kroz usisni kolektor, pri većini režima rada motora, učestvuje sa preko 90% u ukupnom masenom protoku produkata sagorevanja kroz izduvni sistem. Otuda, pretpostavka o proporcionalnosti potrebnog masenog EGR protoka sa masenim protokom kroz usisni kolektor može se smatrati realnom i zadovoljavajućom za potrebe istraživanja u ovom radu.

## 5.2 Struktura simulatora neispravnosti

Simulator neispravnosti predstavlja jednu kombinaciju Matlab programa i Simulink modela, koji funkcionišu u sprezi i na taj način obezbeđuju funkcionisanje kompletnog simulatora. Šema simulatora neispravnosti, sa njegovim ulaznim i izlaznim veličinama, prikazana je na *Sl. 5.1*. Simulink model simulatora neispravnosti predstavlja nadograđeni Hendriksov generički osrednjeni model motora, kojem su dodate neke nove karakteristike, u skladu sa potrebama dijagnostičkog modela. Tako dobijenom novom Simulink modelu dat je naziv ***NSimulator***. Matlab program simulatora neispravnosti kompletno upravlja ***NSimulator***-om: obezbeđuje mu inicijalne podatke, priprema podatke za simuliranje različitih neispravnosti, pokreće ga, preuzima podatke dobijene simulacijom i na kraju

obrađuje i prikazuje rezultate simuliranja neispravnosti. Ovaj program ima naziv ***NSimulatorMenadzer***.



Sl. 5.1 Struktura simulatora neispravnosti, sa ulaznim i izlaznim veličinama

Kako se vidi na Sl. 5.1, izlazne veličine simulatora neispravnosti su broj obrtaja motora  $n$ , pritisak u usisnom kolektoru  $p_i$ , temperatura u usisnom kolektoru  $T_i$ , i maseni protok vazduha pored leptira  $\dot{m}_{at}$ . Postoji više razloga zašto su baš ove veličine odabrane da budu izlaz iz simulatora. Prvi razlog je taj, što su njihove vrednosti dostupne elektronskoj upravljačkoj jedinici u toku rada motora, jer se mere pomoću odgovarajućih senzora. Drugi razlog je što se pretpostavlja da razmatrane neispravnosti u usisnom sistemu mogu da imaju zapažen uticaj na vrednosti  $n$ ,  $p_i$ ,  $T_i$  i  $\dot{m}_{at}$ , što je od esencijalne važnosti za dijagnostikovanje tih neispravnosti. Dalje, vrednosti izlaznih veličina simulatora umnogome zavise od režima rada motora, što je takođe veoma važno sa aspekta dijagnostike. I na kraju, postoji i jedan praktičan razlog, a to je da Hendriksov osrednjeni model motora ima upravo te izlazne veličine (osim  $\dot{m}_{at}$ ), pa ih je simulator neispravnosti jednostavno nasledio iz Hendriksovog modela.

Osnovne ulazne veličine simulatora neispravnosti su otvor leptira izražen u procentima  $\alpha_{PROC}$ , stepen prenosa u kojem se vozilo kreće  $SP$  i temperatura okoline  $T_a$ . Otvor leptira i stepen prenosa su odabrani jer su to glavne promenljive koje definišu režim rada motora, pa se njihovim variranjem može dobiti dovoljan broj različitih radnih tačaka u kojima će se prikupljati podaci za dijagnostički model. Temperatura okoline je činilac koji znatno manje utiče na režim rada motora, ali je značajan sa aspekta detektovanja neispravnosti vezanih za EGR sistem, kao i sa aspekta pogrešnog očitavanja davača temperature u usisnom kolektoru.

### 5.3 Modelovanje simulatora neispravnosti

U prethodnom odeljku rečeno je da se ***NSimulator*** zasniva na Hendriksovom osrednjenom modelu motora, u koji su ugrađene neke dodatne karakteristike. Neke od njih se direktno odnose na sposobnost simuliranja neispravnosti u usisnom sistemu. Druge predstavljaju unapređenja Hendriksovog modela u smislu povećanja broja veličina, čiji uticaj na rad motora može da se prati, pa indirektno omogućavaju da se neispravnosti simuliraju. U nastavku se opisuje način na koji su te dodatne karakteristike modelovane, a

sve to je implementirano u *N Simulator*-u. Najvažniji moduli *N Simulator*-a, koji predstavljaju novost u odnosu na izvorni Hendriksov model, prikazani su u prilogu P1.

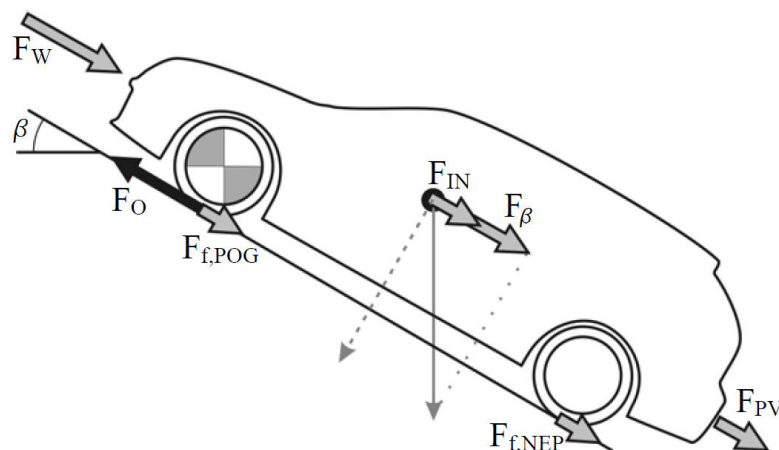
### 5.3.1 Modelovanje snage potrebne za savladavanje otpora na zamajcu

U izvornom Hendriksovom modelu u podsystemu dinamike kolenastog vratila (odjeljak 4.3.3) pretpostavljeno je da je snaga potrebna za savladavanje otpora na zamajcu proporcionalna trećem stepenu broja obrtaja motora, prema izrazu (4.23). Međutim, u tom izrazu nisu sadržane nikakve informacije o vozilu koje motor pogoni, a cilj ovog rada je razvoj dijagnostičkog modela koji bi mogao da pomogne u realizaciji dijagnostičkog sistema na motornom vozilu. Zato je neophodno naći sveobuhvatniji izraz za određivanje snage potrebne za savladavanje otpora na zamajcu, koji bi sadržao veći broj parametara koji utiču na veličinu tih otpora. Da bi se odredili otpori na zamajcu koje motor mora da savlada, najpre treba odrediti otpore kretanja vozila.

Pri određivanju otpora kretanja vozila, uzete su u obzir osnovne pretpostavke pri modelovanu vozila u uzdužnoj dinamici [105]:

- zanemarena su sva pomeranja u pravcima normalnim na pravac kretanja vozila,
- zanemarene su sve pobude na oscilacije i vibracije,
- zanemarene su sve vrste deformacija,
- vozilo se kreće translatorno pravolinijski po idealno ravnoj podlozi,
- dejstvo svih sila i momenata je simetrično u odnosu na središnju uzdužnu ravan vozila,
- vozilo se posmatra samo u uzdužnoj ravni i
- sile na pojedinim točkovima svode se na osovine.

Sile koje deluju na vozilo u pravcu kretanja, na primeru jednog vozila sa pogonom na prednjoj osovini, prikazane su na Sl. 5.2. To su sila otpora kotrljanja pogonske osovine  $F_{f,POG}$ , sila otpora kotrljanja nepogonske osovine  $F_{f,NEP}$ , sila otpora vazduha  $F_W$ , sila otpora uspona  $F_\beta$ , sila otpora inercije  $F_{IN}$  i sila otpora priključnog vozila  $F_{PV}$ , ali će u daljem tekstu sile  $F_{f,POG}$  i  $F_{f,NEP}$  biti objedinjene u jednu, ukupnu silu otpora kotrljanja za vozilo  $F_f$ . Osim sila otpora, na slici je ucrtana i obimna sila na točku  $F_O$ , ali ona ovde neće biti razmatrana, jer je u ovom slučaju cilj da se odrede otpori kretanja.



Sl. 5.2 Sile koje deluju na vozilo u pravcu kretanja [105]

Međutim, pošto je tema istraživanja sužena na ustaljene režime rada motora, ovde će važiti pretpostavka da je brzina kretanja vozila konstantna, što znači da je, u tom slučaju, sila otpora inercije jednaka nuli. Pored toga, smatraće se da posmatrano vozilo ne vuče priključno vozilo, pa je i  $F_{PV} = 0$ . Imajući u vidu opštu jednačinu uzdužne dinamike vozila,

kao i navedene pretpostavke, može se napisati da je ukupna sila otpora vozila  $F_{\text{otp}}$  jednaka obimnoj sili na točku  $F_0$ , [106] i [107],

$$F_{\text{otp}} = F_0 = F_f + F_W + F_\beta . \quad (5.1)$$

To znači da će u obzir biti uzeti otpori kotrljanja, vazduha i uspona.

Veličina sile otpora kotrljanja direktno je proporcionalna vertikalnom opterećenju koje točkove vozila pritiska uz podlogu, a određuje se prema izrazu [106],

$$F_f = f(v) \cdot m_V \cdot g \cdot \cos \beta \quad (5.2)$$

gde je:

$m_V$  - masa vozila,

$g$  - ubrzanje sile teže,

$\beta$  - ugao uzdužnog nagiba podloge i

$f(v)$  - koeficijent otpora kotrljanja

Za određivanje koeficijenta otpora kotrljanja usvojen je jedan od mnogih, u literaturi korišćenih empirijskih izraza [108]

$$f(v) = f_0 + 5.42 \cdot 10^{-6} \cdot v + 1.05 \cdot 10^{-11} \cdot v^4 \quad (5.3)$$

gde je  $v$  brzina kretanja vozila u [km/h], a  $f_0$  je statička vrednost koeficijenta otpora kotrljanja i, za dobru asfaltnu podlogu ima vrednost 0.01.

Sila otpora vazduha zavisi od gustine vazduha  $\rho$ , koeficijenta otpora vazduha  $c_W$ , veličine čeone površine vozila  $A$ , brzine kretanja vozila  $v$  i brzine vetra  $v_{\text{wind}}$  u pravcu kretanja vozila, prema izrazu [107]

$$F_W = \frac{1}{2} \cdot c_W \cdot A \cdot \rho \cdot (v + v_{\text{wind}})^2 \quad (5.4)$$

gde je brzina vetra pozitivna, ako vetar ima smer suprotan kretanju vozila, a negativna, ako su im smerovi isti.

Intenzitet sile otpora uspona određuje se iz jednostavnog izraza

$$F_\beta = m_V \cdot g \cdot \sin \beta, \quad (5.5)$$

čime postaje poznata i poslednja komponenta ukupne sile otpora vozila iz jednačine (5.1).

Množenjem izraza (5.1) sa brzinom kretanja vozila, dobija se jednačina koja povezuje potrebnu snagu na točku  $P_T$ , sa snagama potrebnim za savladavanje svakog pojedinačnog otpora pri datoj brzini kretanja,

$$P_T = P_f + P_W + P_\beta . \quad (5.6)$$

Ako se gubici koji postoje pri prenosu snage od motora do pogonskih točkova uzmu u obzir preko koeficijenta iskorišćenja transmisije  $\eta_{\text{TR}}$ , kombinovanjem jednačina (5.2)-(5.6) dobija se izraz za snagu potrebnu za savladavanje otpora na zamajcu motora

$$P_b(v) = \frac{[(f_0 + 5.42 \cdot 10^{-6} \cdot v + 1.05 \cdot 10^{-11} \cdot v^4) \cdot m_V \cdot g \cdot \cos \beta + \frac{1}{2} \cdot c_W \cdot A \cdot \rho \cdot (v + v_{\text{wind}})^2 + m_V \cdot g \cdot \sin \beta] \cdot v}{\eta_{\text{TR}}} \quad (5.7)$$

u zavisnosti od brzine kretanja vozila.

Međutim, u originalnom Hendriksovom modelu, u podsistemu dinamike kolenastog vratila, potrebna je zavisnost  $P_b(n)$ . Ova zavisnost se može dobiti, ako se u izraz (5.7) uvrsti veza između brzine kretanja vozila i broja obrtaja:

$$v \left[ \frac{\text{km}}{\text{h}} \right] = 3600 \cdot r_D \cdot \frac{\pi \cdot n}{30 \cdot i_M \cdot i_{\text{GP}}} \quad (5.8)$$

gde je:

$r_D$  - dinamički radijus točka [m],

$i_M$  – prenosni odnos menjačkog prenosnika i

$i_{GP}$  – prenosni odnos glavnog prenosnika.

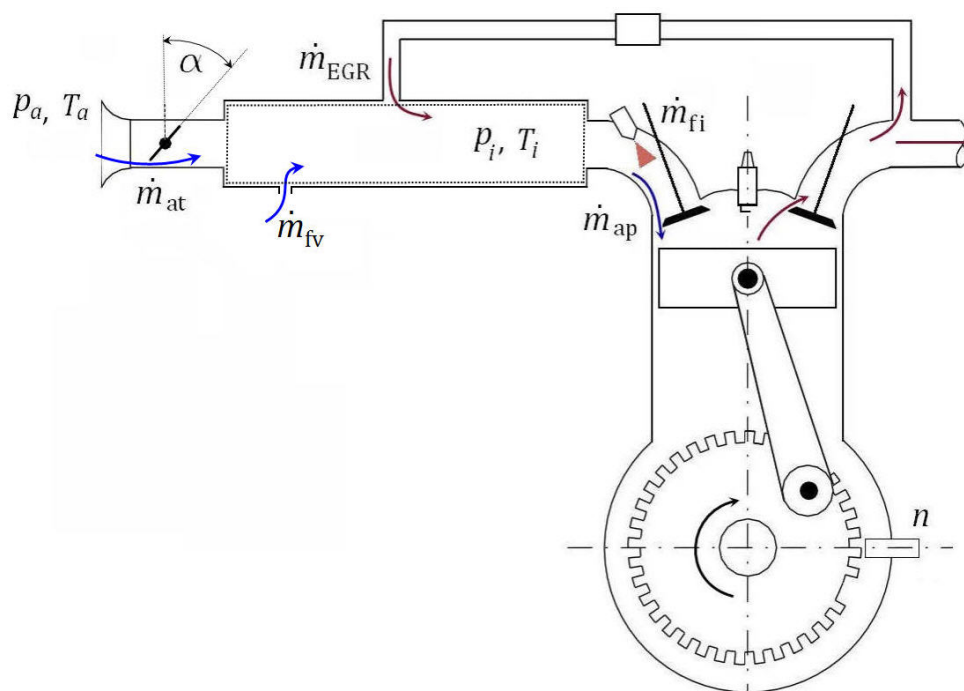
Zbog velike dužine konačnog novoizvedenog izraza  $P_b(n)$ , njegov prikaz će ovde biti izostavljen. Međutim, ugrađivanjem novog izraza  $P_b(n)$  u jednačinu (4.20), a time i u **NSimulator**, dobija se mogućnost simuliranja uticaja različitih parametara kretanja vozila kao što su stepen prenosa, uzdužni nagib puta, brzina vetra i masa vozila na modelovane veličine: broj obrtaja motora, pritisak i temperaturu u usisnom kolektoru.

### 5.3.2 Modelovanje uticaja nezaptivenosti usisnog sistema

Nezaptivenošću usisnog sistema ovde će se smatrati postojanje nekog nepredviđenog otvora u delu usisnog sistema koji se nalazi iza leptira, kada se posmatra u smeru strujanja vazduha. Do pojave takvog otvora najčešće dolazi kada neko vakuumsko crevo, povezano sa usisnim sistemom, spadne sa svog uvodnika ili kada se ošteti neko od tih creva ili usled oštećenja same usisne cevi. Ukoliko se to desi, kod usisnih (atmosferskih) motora će doći do ustujavanja vazduha iz okolne sredine u usisni sistem motora, jer je pritisak u usisnom sistemu uvek niži od okolnog atmosferskog pritiska. Kod prehranjivanih motora može doći do strujanja vazduha kroz nepredviđeni otvor u bilo kom smeru (iz okolne sredine u usisni sistem ili obrnuto, iz usisnog sistema u okolnu sredinu), u zavisnosti od režima rada, ali ti motori ovde nisu tema istraživanja.

Na Sl. 5.3 ilustrovana je situacija, kada postoji nezaptivenost u usisnom sistemu jednog atmosferskog motora. Za vazduh, koji u tom slučaju ustujava u usisni sistem kroz nepredviđeni otvor, odomaćen je izraz „fals“ -vazduh, pa će, zbog jednostavnosti, i ovde biti korišćen taj termin.

U novonastaloj situaciji ne važe više jednačine (4.7) i (4.19) za određivanje pritiska i temperature u usisnom kolektoru. Zato je neophodno izvesti nove izraze, koji će važiti pri postojanju još jednog masenog protoka koji ulazi u usisni kolektor. To izvođenje biće prikazano u daljem tekstu, a biće primenjeni prvi princip termodinamike za otvoreni sistem i zakon o održanju mase. Taj otvoreni sistem je u ovom slučaju usisni kolektor, a on ujedno predstavlja i kontrolnu zapreminu, čije granice su na Sl. 5.3 obeležene isprekidanom linijom.



Sl. 5.3 Maseni protoci kroz usisni sistem pri postojanju neželjenog otvora iza leptira

### 5.3.2.1 Uticaj na temperaturu u usisnom kolektoru

Primena zakona o održanju mase na označenu kontrolnu zapreminu, dovodi do sledećeg izraza

$$\dot{m}_i = \dot{m}_{at} + \dot{m}_{EGR} - \dot{m}_{ap} + \dot{m}_{fv} \quad (5.9)$$

gde je

$\dot{m}_i$  - promena mase gasa u usisnom kolektoru sa vremenom,

$\dot{m}_{at}$  - maseni protok vazduha oko leptira,

$\dot{m}_{EGR}$  - maseni protok recirkuliranih izduvni gasova,

$\dot{m}_{ap}$  - maseni protok kroz usisne kanale i

$\dot{m}_{fv}$  - maseni protok „falš“-vazduha.

Dalje, da bi se dobio upotrebljiv oblik prvog zakona termodinamike za posmatranu kontrolnu zapreminu, neophodno je uvesti određene pretpostavke. Pri tome su neke pretpostavke nasleđene iz Hendriksovog osrednjenog modela motora, a uvedene su i neke nove. Zbog kompaktnosti teksta, sve one će ovde biti navedene:

- nema razmene toplote između kontrolne zapremine i okoline,
- ne vrši se nikakav rad nad kontrolnom zapreminom, osim rada koji je posledica protoka materije (rad strujanja)
- zanemaruju se promene kinetičke i potencijalne energije ulaznih i izlaznih protoka kontrolne zapremine,
- i vazduh i gasovi iz sistema recirkulacije smatraju se idealnim gasom,

Uvedene pretpostavke u nekoj meri idealizuju procese strujanja koji se dešavaju u usisnom kolektoru, ali bez obzira na to, primena uprošćenog oblika prvog zakona termodinamike na takav sistem predstavlja prilično dobru aproksimaciju [109].

Imajući u vidu sve ove pretpostavke, prvi zakon termodinamike za posmatrani sistem najpogodnije je napisati u formi entalpija [109]

$$\frac{dU}{dt} = \dot{m}_{at} \cdot h_a + \dot{m}_{EGR} \cdot h_{EGR} + \dot{m}_{fv} \cdot h_a - \dot{m}_{ap} \cdot h_i \quad (5.10)$$

gde su:

$U$  - unutrašnja energija posmatranog sistema,

$h_a$  - specifična entalpija vazduha,

$h_{EGR}$  - specifična entalpija recirkuliranih izduvni gasova i

$h_i$  - specifična entalpija gasa koji izlazi iz kontrolne zapremine.

Unutrašnja energija posmatranog sistema može se izraziti kao

$$U = m_i \cdot c_v \cdot T_i \quad (5.11)$$

gde je  $c_v$  specična toplota pri konstantnoj zapremini.

Diferenciranjem po vremenu jednačine (5.11) i uvrštavanjem tako dobijenog izraza za  $\frac{dU}{dt}$  u jednačinu (5.10), dobija se

$$\dot{m}_i \cdot c_v \cdot T_i + m_i \cdot c_v \cdot \dot{T}_i = \dot{m}_{at} \cdot h_a + \dot{m}_{EGR} \cdot h_{EGR} + \dot{m}_{fv} \cdot h_a - \dot{m}_{ap} \cdot h_i \quad (5.12)$$

Sada je cilj osloboditi se specifičnih entalpija i specifičnih toplota, kako bi se dobila upotrebljiva jednačina, pomoću koje se može odrediti temperatura u usisnom kolektoru. Poznato je da postoji sledeća veza između specifične entalpije i temperature neke materije

$$h = c_p \cdot T \quad (5.13)$$

gde je  $c_p$  specifična toplota pri konstantnom pritisku. Isto tako, zna se da su specifične toplote  $c_p$  i  $c_v$  povezane sledećim izrazom:

$$\frac{c_p}{c_v} = \kappa. \quad (5.14)$$

Kombinujući jednačine (5.12)-(5.14) i rešavanjem jednačine (5.12) po  $\dot{T}_i$ , dobija se

$$\dot{T}_i = \frac{1}{m_i} \cdot (\dot{m}_{at} \cdot \kappa \cdot T_a + \dot{m}_{EGR} \cdot \kappa \cdot T_{EGR} + \dot{m}_{fv} \cdot \kappa \cdot T_a - \dot{m}_{ap} \cdot \kappa \cdot T_i - \dot{m}_i \cdot T_i). \quad (5.15)$$

Korišćenjem jednačine stanja idealnog gasa, može se odrediti masa gasa u usisnom kolektoru

$$m_i = \frac{p_i \cdot V_i}{R \cdot T_i} \quad (5.16)$$

gde je

$R$  - gasna konstanta vazduha,

$V_i$  - zapremina usisnog kolektora.

Ovde treba napomenuti da je u jednačini (5.16) korišćena gasna konstanta vazduha, jer veći deo materije koja ispunjava usisni kolektor, čini upravo vazduh.

Ako se umesto  $m_i$  i  $\dot{m}_i$  u jednačini (5.15) uvrste izrazi (5.9) i (5.16), i izvrši određeno sređivanje, dobija se da je

$$\dot{T}_i = \frac{R \cdot T_i}{p_i \cdot V_i} \cdot [\dot{m}_{at}(\kappa T_a - T_i) + \dot{m}_{EGR}(\kappa T_{EGR} - T_i) + \dot{m}_{fv}(\kappa T_a - T_i) - \dot{m}_{ap}(\kappa - 1)T_i]. \quad (5.17)$$

Maseni protoci  $\dot{m}_{at}$ ,  $\dot{m}_{EGR}$  i  $\dot{m}_{ap}$  su poznati iz izvornog Hendriksovog modela, a ovde još preostaje da se odredi maseni protok „falš“-vazduha.

Strujanje vazduha kroz neželjeni otvor u usisnom sistemu modelovaće se kao jednodimenzionalno izentropsko adijabatsko i stacionarno strujanje kompresibilnog fluida kroz suženje. Imajući u vidu ova ograničenja, kao i ranije uvedenu pretpostavku da je vazduh idealan gas, ovde će važiti sledeće relacije [97]:

- energijska jednačina stacionarnog strujanja

$$c_p \cdot T + \frac{1}{2} \cdot v_{otv}^2 = c_p \cdot T_a \quad (5.18)$$

- izentropske relacije

$$\left(\frac{p}{p_a}\right) = \left(\frac{T}{T_a}\right)^{\frac{\kappa}{\kappa-1}} = \left(\frac{\rho}{\rho_a}\right)^{\kappa} \quad (5.19)$$

- izraz za brzinu zvuka u vazduhu

$$c = \sqrt{\kappa \cdot R \cdot T} \quad (5.20)$$

- jednačina stanja idealnog gasa u obliku

$$\rho = \frac{p}{R \cdot T} \quad (5.21)$$

gde su:

$p_a$ ,  $T_a$  i  $\rho_a$  - pritisak, temperatura i gustina vazduha ispred suženja,

$p$ ,  $T$  i  $\rho$  - pritisak, temperatura i gustina vazduha neposredno iza suženja i

$v_{otv}$  - brzina strujanja vazduha kroz suženje.

Izvorna jednačina, iz koje se određuje maseni protok kroz otvor površine  $A_{fv}$ , je jednačina kontinuiteta oblika

$$\dot{m} = \mu \cdot \rho \cdot A_{fv} \cdot v_{otv} \quad (5.22)$$

gde je  $\mu$  koeficijent protoka kroz otvor. Zbog lakšeg definisanja veličine otvora kroz koji struji „falš“-vazduh, pretpostaviće se da je otvor kružnog oblika, prečnika  $d_{fv}$ , i u sledećim jednačinama figurisaće ta oznaka umesto  $A_{fv}$ .

Primenom jednačine (5.22) na konkretan slučaj strujanja „falš“-vazduha kroz neželjeni otvor u usisnom sistemu, uz korišćenje izraza (5.18)-(5.21), kao i osnovnih relacija

$$R = c_p - c_v \quad (5.23)$$

i

$$\kappa = \frac{c_p}{c_v} \quad (5.24)$$

dobija se izraz za određivanje masenog protoka „falš“-vazduha, koji posle sređivanja ima sledeći oblik:

$$\dot{m}_{fv} = \frac{\mu \cdot d_{fv}^2 \cdot \pi \cdot p_a}{4 \cdot \sqrt{R \cdot T_a}} \cdot \left(\frac{p_i}{p_a}\right)^{\frac{1}{\kappa}} \cdot \sqrt{\frac{2 \cdot \kappa}{\kappa - 1} \cdot \left[1 - \left(\frac{p_i}{p_a}\right)^{\frac{\kappa-1}{\kappa}}\right]} \quad (5.25)$$

Treba imati na umu da je ovde u pitanju strujanje kompresibilnog fluida, za koje je karakteristična pojava prigušenja protoka, kada brzina strujanja dostigne vrednost brzine zvuka. O tome je već nešto rečeno u odeljku 4.3.2, a ovde će biti data dopuna. Dakle, protok vazduha kroz neželjeni otvor raste sa povećanjem brzine strujanja, a brzina strujanja raste sa smanjenjem pritiska u usisnom kolektoru. Međutim, protok će se povećavati samo dok se brzina strujanja ne izjednači sa brzinom zvuka. To se dešava pri nekoj kritičnoj vrednosti pritiska u usisnom kolektoru koja je određena izrazom

$$(p_i)_{krit} = p_a \cdot \left(\frac{2}{\kappa+1}\right)^{\frac{\kappa}{\kappa-1}} \quad (5.26)$$

koji se jednostavno dobija kombinovanjem jednačina (5.18)-(5.20) i (5.23)-(5.24) i uzimanjem u obzir činjenice, da je brzina strujanja kroz neželjeni otvor jednaka brzini zvuka. Dalje snižavanje pritiska u usisnom kolektoru ispod vrednosti  $(p_i)_{krit}$ , ne dovodi ni do porasta brzine strujanja, ni do porasta protoka, što znači da je dostignuta maksimalna vrednost protoka. Ova vrednost se dobija uvrštavanjem jednačine (5.26) u (5.25) i posle sređivanja ima oblik

$$(\dot{m}_{fv})_{max} = \frac{\mu \cdot d_{fv}^2 \cdot \pi \cdot p_a}{4 \cdot \sqrt{R \cdot T_a}} \cdot \sqrt{\kappa \cdot \left(\frac{2}{\kappa+1}\right)^{\frac{\kappa+1}{\kappa-1}}} \quad (5.27)$$

Na osnovu rečenog, konačni izraz za maseni protok „falš“-vazduha biće

$$\dot{m}_{fv} = \begin{cases} \frac{\mu \cdot d_{fv}^2 \cdot \pi \cdot p_a}{4 \cdot \sqrt{R \cdot T_a}} \cdot \left(\frac{p_i}{p_a}\right)^{\frac{1}{\kappa}} \cdot \sqrt{\frac{2 \cdot \kappa}{\kappa - 1} \cdot \left[1 - \left(\frac{p_i}{p_a}\right)^{\frac{\kappa-1}{\kappa}}\right]} & \text{za } p_i > (p_i)_{krit} \\ \frac{\mu \cdot d_{fv}^2 \cdot \pi \cdot p_a}{4 \cdot \sqrt{R \cdot T_a}} \cdot \sqrt{\kappa \cdot \left(\frac{2}{\kappa+1}\right)^{\frac{\kappa+1}{\kappa-1}}} & \text{za } p_i \leq (p_i)_{krit} \end{cases} \quad (5.28)$$

a njegovim uvrštavanjem u jednačinu (5.17) biće modelovan uticaj strujanja „falš“-vazduha u usisni kolektor na temperaturu u usisnom kolektoru. Osim na temperaturu, ulazak „falš“-vazduha u usisni sistem motora, uticaće i na pritisak u usisnom kolektoru. U nastavku će biti modelovan i taj uticaj.



### 5.3.2.2 Uticaj na pritisak u usisnom kolektoru

Polazi se od primene jednačine stanja idealnog gasa na usisni kolektor, iz koje se može izraziti pritisak

$$p_i = \frac{R}{V_i} \cdot m_i \cdot T_i \quad (5.29)$$

čijim diferenciranjem po vremenu se dobija

$$\dot{p}_i = \frac{R}{V_i} \cdot (\dot{m}_i \cdot T_i + m_i \cdot \dot{T}_i). \quad (5.30)$$

Ako se iz jednačine (5.29) izrazi masa  $m_i$ , i uzmu u obzir jednačine (5.9) i (5.17), jednačina (5.30) posle određenog sređivanja postaje

$$\dot{p}_i = \frac{\kappa \cdot R}{V_i} \cdot \left( (\dot{m}_{at} + \dot{m}_{fv}) \cdot T_a + \dot{m}_{EGR} \cdot T_{EGR} - \dot{m}_{ap} \cdot T_i \right). \quad (5.31)$$

Ugrađivanjem jednačina (5.17) i (5.31) u simulink model simulatora neispravnosti može se oponašati direktan uticaj „falš“-vazduha na temperaturu i pritisak u usisnom kolektoru. Međutim, tu je još i indirektni uticaj, jer će se svakako i doziranje goriva poremetiti. Modelovanje doziranja goriva u slučaju postojanja neželjenog otvora u usisnom sistemu biće opisano kasnije, kada budu poznati načini modelovanja ostalih neispravnosti u usisnom sistemu.

### 5.3.3 Modelovanje pogrešnog očitavanja senzora pritiska u usisnom kolektoru

Pogrešno očitavanje senzora pritiska u usisnom kolektoru različito se odražava na rad motora sa izračunavanjem masenog protoka vazduha (bez masenog protokomera) i motora sa merenjem masenog protoka vazduha (sa masenim protokomerom). Zato će ova dva slučaja biti razdvojena. Pre razdvajanja ova dva slučaja, treba otkloniti i jednu eventualnu terminološku zabunu, u vezi sa pojmom „maseni protok vazduha“.

Kada se kod motora sa merenjem protoka vazduha kaže „maseni protok vazduha“ misli se na maseni protok vazduha kroz protokomer, tj. na vazduh koji ulazi u usisni kolektor struajući oko leptira, ako nije drugačije naznačeno. Ovaj protok je označen sa  $\dot{m}_{at}$  (Sl. 5.3).

Sa druge strane, kod motora sa računanjem masenog protoka vazduha, pojam „maseni protok vazduha“ označava ukupan protok kroz usisni kolektor, ako drugačije nije naznačeno. Ravnopravno će se koristiti još i pojmovi „maseni protok kroz usisni kolektor“, kao i ranije korišćeni „maseni protok kroz usisne kanale“, jer se ove dve veličine smatraju jednakima. Oznaka koja važi za svaki od ovih pojmova je  $\dot{m}_{ap}$  (Sl. 5.3). Ovde treba još napomenuti da je u masenom protoku kroz usisni kolektor, osim vazduha, sadržan i maseni protok recirkulisanih izduvnih gasova, kada je EGR sistem aktivan.

#### 5.3.3.1 Motori bez masenog protokomera vazduha

Signal davača pritiska u usisnom kolektoru ima mnogo značajniju ulogu kod motora koji nemaju maseni protokomer vazduha. Kod takvih motora, maseni protok vazduha se izračunava, prema tzv. „speed-density“ formuli. U Hendriksovom modelu, u tu svrhu služi modifikovana „speed-density“ formula koju reprezentuje jednačina (4.18) i koja će, zbog lakšeg praćenja, ovde biti ponovo navedena

$$\dot{m}_{ap} = \frac{V_d}{120 \cdot R \cdot T_i} \cdot (s_i \cdot p_i - y_i) \cdot n. \quad (5.32)$$

Iz jednačine se vidi da je pritisak u usisnom kolektoru jedan od bitnijih činilaca koji utiču na određivanje masenog protoka kroz usisne kanale  $\dot{m}_{ap}$ . Otuda će se pogrešno očitavanje pritiska odraziti na izračunatu veličinu masenog protoka kroz usisni kolektor, koja se koristi za doziranje goriva. Zbog toga je, u svrhu simuliranja ove neispravnosti, neophodno da umesto  $p_i$ , u modulu za doziranje goriva u simulatoru neispravnosti figuriše izraz  $(G_{pgain} \cdot p_i + G_{poffset})$ . Sa druge strane, na svim drugim mestima u simulatoru potrebno je da figuriše ispravna vrednost  $p_i$ , da bi on funkcionisao kako treba. To je izvedeno tako što je ispravna vrednost pritiska  $p_i$  zadržana kao takva u celom modelu, a pogrešno izračunata vrednost  $(\dot{m}_{ap})_{pGreš}$  zbog greške senzora pritiska, određuje se u modulu za doziranje goriva prema jednačini [110]

$$(\dot{m}_{ap})_{pGreš} = \frac{\dot{m}_{ap}}{s_i \cdot p_i - y_i} \cdot [s_i \cdot (G_{pgain} \cdot p_i + G_{poffset}) - y_i]. \quad (5.33)$$

Dakle, jednostavno je ispravna vrednost  $\dot{m}_{ap}$  podeljena sa članom  $(s_i \cdot p_i - y_i)$ , u kojem figuriše ispravna vrednost pritiska, a potom pomnožena sa istim tim članom  $[s_i \cdot (G_{pgain} \cdot p_i + G_{poffset}) - y_i]$ , u kojem figuriše pogrešno očitana vrednost pritiska. Veličine  $G_{pgain}$  i  $G_{poffset}$  označavaju grešku pojačanja i grešku ofseta davača pritiska, respektivno. Na taj način je omogućeno da se pomoću simulatora neispravnosti simulira uticaj ove dve vrste grešaka očitavanja senzora pritiska na računanje veličine  $\dot{m}_{ap}$ . Pogrešno izračunat maseni protok vazduha utiče na doziranje goriva, prema izrazu (4.4), o čemu će biti reči kasnije, u posebnom odeljku.

### 5.3.3.2 Motori sa masenim protokomerom vazduha

Kod motora, koji poseduju maseni protokomer vazduha, davač pritiska u usisnom kolektoru nema tako značajnu ulogu, odnosno direktno ne utiče na doziranje goriva. Neki motori sa masenim protokomerom vazduha čak ni nemaju senzor pritiska u usisnom kolektoru, ali postoje mnogi koji ga imaju. U tom slučaju, on uglavnom služi u dijagnostičke svrhe, za otkrivanje neispravnosti na drugim senzorima i sistemima (najčešće na EGR ventilima) ili kao rezervna opcija za određivanje opterećenja motora, kada otkaže maseni protokomer vazduha [111]. Zbog toga će pogrešno očitavanje senzora pritiska ovde biti simulirano jednostavno pomoću izraza

$$(p_i)_{greš} = G_{pgain} \cdot p_i + G_{poffset} \quad (5.34)$$

gde je  $(p_i)_{greš}$  simulirana pogrešno očitana vrednost pritiska u usisnom kolektoru. Ova greška neće imati uticaja na doziranje goriva, jer je kod razmatrane vrste motora, za ovu svrhu dominantan signal protokomera vazduha.

### 5.3.4 Modelovanje pogrešnog očitavanja senzora temperature u usisnom kolektoru

Pri modelovanju pogrešnog očitavanja senzora temperature u usisnom kolektoru, od interesa su samo motori kod kojih se maseni protok vazduha izračunava. Naime, uglavnom se samo kod ovih motora meri temperatura u usisnom kolektoru. Razlog je taj što gustina vazduha, koji struji kroz usisni kolektor, osim od pritiska, značajno zavisi i od temperature u usisnom kolektoru, što se vidi i iz jednačine (5.32). Otuda, pogrešno očitavanje temperature u usisnom kolektoru dovodi i do pogrešnog izračunavanja masenog protoka vazduha kroz usisni kolektor (usisne kanale). Iz istih razloga kao kod pogrešnog očitavanja senzora pritiska, u modelu će svugde figurisati ispravna vrednost temperature  $T_i$ . Međutim,

u modulu za doziranje goriva simuliraće se izračunata pogrešna vrednost masenog protoka kroz usisni kolektor  $(\dot{m}_{ap})_{T_{Greš}}$  zbog greške senzora temperature, na osnovu izraza [112]

$$(\dot{m}_{ap})_{T_{Greš}} = \frac{\dot{m}_{ap}}{T_i} \cdot (G_{T_{gain}} \cdot T_i + G_{T_{offset}}) . \quad (5.35)$$

Ovde je, takođe, primenjen princip da se ispravna vrednost  $\dot{m}_{ap}$  podeli sa ispravnom vrednošću temperature u usisnom kolektoru, pa onda pomnoži sa njenom pogrešno očitanom vrednošću, kako bi se simulirala neispravna vrednost masenog protoka. Koeficijenti  $G_{T_{gain}}$  i  $G_{T_{offset}}$  označavaju grešku pojačanja i grešku ofseta senzora temperature, respektivno. Izborom različitih vrednosti nekog od ovih koeficijenata, mogu se simulirati različiti intenziteti ove dve vrste grešaka.

Motori sa masenim protokomerom vazduha uglavnom ne poseduju senzor temperature u usisnom kolektoru, već imaju senzor temperature vazduha koji je vrlo često integrisan u samom protokomeru. Ovaj senzor treba da omogući da signal masenog protokomera vazduha ne zavisi od spoljašnje temperature. Greške ovakvih senzora temperature vazduha ovde nisu od interesa i neće se posebno razmatrati. One se mogu tretirati kao greške očitavanja masenog protokomera vazduha, jer uglavnom čine njegov sastavni deo.

### 5.3.5 Modelovanje pogrešnog očitavanja senzora masenog protoka vazduha

Posmatrano sa aspekta doziranja goriva, kod motora sa merenjem masenog protoka vazduha najvažniji je upravo signal senzora pomoću kojeg se to merenje ostvaruje. Zbog toga će se pogrešno očitavanje senzora masenog protoka vazduha, odraziti i na kvalitet obrazovane smeše u cilindru. Kako je objašnjeno na početku odeljka 5.3.3, veličina koja se ovde meri je maseni protok vazduha koji ulazi u usisni kolektor,  $\dot{m}_{at}$ . Za razliku od motora bez masenog protokomera vazduha, kod kojih se protok vazduha određuje indirektno, ovde se to čini direktno, pa je izraz za modelovanje pogrešnog očitavanja masenog protokomera vazduha  $(\dot{m}_{at})_{MAFGreš}$  vrlo jednostavan:

$$(\dot{m}_{at})_{MAFGreš} = G_{MAF_{gain}} \cdot \dot{m}_{at} + G_{MAF_{offset}} . \quad (5.36)$$

Koeficijenti  $G_{MAF_{gain}}$  i  $G_{MAF_{offset}}$  označavaju greške pojačanja i ofseta masenog protokomera vazduha, respektivno, a pomoću njih se mogu simulirati različiti intenziteti ove dve vrste grešaka.

### 5.3.6 Modelovanje umanjenog EGR protoka

Pri funkcionisanju ispravnog motora, i u režimu otvorene i u režimu zatvorene sprege sa lambda-sondom, maseni EGR protok je modelovan tako, da je proporcionalan registrovanom masenom protoku vazduha kroz usisni kolektor. Taj koeficijent proporcionalnosti označen je sa  $k_{EGR_{max}}$ , jer se na srednjim opterećenjima i brojevima obrtaja, najbolji rezultati umanjena sadržaja azotnih oksida u izduvnim gasovima postižu uglavnom pri maksimalno dozvoljenim vrednostima EGR protoka. Zbog proporcionalnosti sa masenim protokom vazduha i činjenice da se taj protok određuje na različite načine kod motora sa masenim protokomerom i kod motora bez protokomera, maksimalni ili željeni maseni EGR protok će biti modelovan posebno za svaku posmatranu vrstu motora.

Kod motora sa masnim protokomerom, izraz za određivanje maksimalnog EGR protoka glasi

$$\dot{m}_{EGR_{max}} = k_{EGR_{max}} \cdot \dot{m}_{at} \quad (5.37)$$

dok u slučaju motora bez protokomera, ima oblik

$$\dot{m}_{EGRmax} = \frac{k_{EGRmax}}{k_{EGRmax} + 1} \cdot \dot{m}_{ap} \quad (5.38)$$

Pri radu motora bez neispravnosti, ove dve jednačine daju iste rezultate, ali kada postoje neke neispravnosti u usisnom sistemu, rezultati će se razlikovati. Ovo naročito dolazi do izražaja pri simuliranju ulaska „falš“-vazduha u usisni sistem, jer veličina  $\dot{m}_{ap}$  u sebi sadrži maseni protok „falš“-vazduha, što se ne može reći za veličinu  $\dot{m}_{at}$ .

Do umanjena željenog protoka može doći usled zaglavlivanja EGR ventila u nekom položaju ili usled zagušenja EGR voda čvrstim komponentama produkata sagorevanja. Umanjeni protok modelovan je množenjem željenog (maksimalnog) EGR protoka koeficijentom  $k_{EGR}$ , što se može napisati kao

$$\dot{m}_{EGR} = k_{EGR} \cdot \dot{m}_{EGRmax} \quad (5.39)$$

pri čemu se izborom koeficijenta  $k_{EGR}$  mogu simulirati različite veličine zagušenja u EGR vodu.

Koeficijentom  $k_{EGRmax}$  može se u simulatoru varirati vrednost željenog protoka, ali je ovde usvojeno da taj koeficijent proporcionalnosti bude 0.2, u skladu sa objašnjenjem datim na kraju odeljka 5.1.

Treba napomenuti da izrazi (5.37)-(5.39) nisu izrazi na osnovu kojih ECU motora računa potreban maseni EGR protok, nego su aproksimacija, koja na zadovoljavajući način, za potrebe simuliranja neispravnosti, opisuje prisustvo recirkulisanih izduvnihi gasova u usisnom sistemu.

### 5.3.7 Modelovanje doziranja goriva u slučaju neispravnosti u usisnom sistemu

Većina prethodno razmotrenih neispravnosti utiče na doziranje goriva, pa je trebalo izvršiti određene promene u originalnom Hendriksovom modelu i u tom segmentu, kako bi se kompletirao *NSimulator*. Pošto se *NSimulator* zasniva na osrednjenom modelu motora, pojam „doziranje goriva“ se ovde ne odnosi na određivanje doze goriva po jednom radnom ciklusu, nego na određivanje masenog protoka ubrizganog goriva, potrebnog za postizanje željenog kvaliteta smeše.

S obzirom na ranije definisane dijagnostičke zadatke koji bi trebalo da se realizuju u ovom radu, treba razlikovati sledeća četiri slučaja:

1. Motor sa masenim protokomerom radi u režimu otvorene sprege sa lambda-sondom,
2. Motor bez protokomera radi u režimu otvorene sprege sa lambda-sondom,
3. Motor sa masenim protokomerom radi u režimu zatvorene sprege sa lambda-sondom,
4. Motor bez protokomera radi u režimu zatvorene sprege sa lambda-sondom.

Međutim, sa aspekta modelovanja simulatora, korekcije sastava smeše u zatvorenoj sprezi sa lambda-sondom ne razlikuju se kod posmatrane dve vrste motora, pa se treći i četvrti slučaj mogu objediniti u jedan.

Bez obzira na to, da li motor radi u zatvorenoj ili otvorenoj sprezi sa lambda-sondom, postoji neka željena vrednost koeficijenta viška vazduha  $\lambda_{zelj}$ , kojoj se teži. Ova vrednost u originalnom Hendriksovom modelu je konstantna i jednaka je jedinici, bez obzira na opterećenje i broj obrtaja motora. Međutim, zbog opštosti, u *NSimulator* ugrađena je mapa (tabela) željenih vrednosti, u zavisnosti od opterećenja i broja obrtaja motora. Konkretno željene vrednosti preuzete su sa stvarnog motora koji je poslužio za eksperimentalni deo

istraživanja, što će biti opisano kasnije, u odeljku o ulaznim podacima simulatora neispravnosti (odeljak 5.4.1).

Sa aspekta otkrivanja i prepoznavanja razmatranih neispravnosti u usisnom sistemu, bitno je da postojanje neke od tih neispravnosti utiče na promenu bar jedne od izlaznih veličina simulatora neispravnosti, a to su broj obrtaja, pritisak u usisnom kolektoru i temperatura u usisnom kolektoru. U stvari, promene ovih veličina samo su manifestacija pojave neispravnosti, a ono što se suštinski dešava sa pojavom neke od ovih neispravnosti, jeste promena razvijene indikatorske snage motora  $P_i$ . Ova snaga se u Hendriksovom modelu, pa i u **NSimulator**-u određuje iz izraza

$$P_i = \eta_i \cdot H_u \cdot \dot{m}_f \quad (5.40)$$

gde su ključne veličine maseni protok goriva kroz usisne kanale  $\dot{m}_f$  i indikatorski stepen iskorišćenja  $\eta_i$  ( $H_u$  je donja toplotna moć goriva i konstantna je). U daljem razmatranju, umesto veličine  $\dot{m}_f$ , koristiće se maseni protok ubrizganog goriva  $\dot{m}_{fi}$ , jer se u ustaljenom režimu rada motora te dve veličine mogu smatrati jednakima.

Veličina  $\dot{m}_{fi}$  je bitna, jer pojava neke neispravnosti dovodi do poremećaja u doziranju goriva, pa će zato doći i do promene u razvijenoj snazi.

Veličina  $\eta_i$ , prema jednačini (4.24), između ostalog, zavisi i od koeficijenta viška vazduha, pa pojava neke neispravnosti može da utiče na razvijenu indikatorsku snagu i preko ovog parametra.

### 5.3.7.1 Motor sa masenim protokomerom, otvorena sprega sa $\lambda$ -sondom

Kod motora sa masenim protokomerom vazduha, potrebni maseni protok ubrizganog goriva  $\dot{m}_{fi}$  određuje se na osnovu izmerenog masenog protoka vazduha  $\dot{m}_{at}$ , iz jednačine

$$\dot{m}_{fi} = \frac{\dot{m}_{at}}{\lambda_{želj} \cdot L_{th}} \quad (5.41)$$

U ovom slučaju na doziranje goriva utiču sledeće neispravnosti:

- pogrešno očitavanje protokomera vazduha,
- ulaz „falš“-vazduha u usisni sistem i
- umanjen EGR protok.

#### Pogrešno očitavanje protokomera vazduha

Usled pogrešnog očitavanja protokomera vazduha, izračunata vrednost potrebnog masenog protoka ubrizganog goriva će biti

$$\dot{m}_{fi} = \frac{(\dot{m}_{at})_{MAFGreš}}{\lambda_{želj} \cdot L_{th}} \quad (5.42)$$

gde je  $(\dot{m}_{at})_{MAFGreš}$  pogrešno očitana vrednost masenog protoka vazduha pored leptira, koja se u modelu simulatora određuje pomoću jednačine (5.36).

Ostvarivanjem ovako izračunatog protoka goriva, dobiće se smeša drugačijeg sastava od željenog. Taj sastav smeše definisan je novom vrednošću koeficijenta viška vazduha, koji će biti nazvan stvarnim koeficijentom viška vazduha,  $\lambda_{stv}$ . Vrednost  $\lambda_{stv}$  određena je izrazom

$$\lambda_{stv} = \frac{\dot{m}_{at}}{\dot{m}_{fi} \cdot L_{th}} \quad (5.43)$$

gde je  $\dot{m}_{at}$  stvarna vrednost masenog protoka vazduha pored leptira, a ne ona koju je pogrešno očitao maseni protokomer vazduha.

## Ulaz „falš“-vazduha u usisni sistem

Kada postoji otvor kroz koji u usisni sistem ulazi neželjeni vazduh, vrednost potrebnog masenog protoka ubrizganog goriva računa se samo na osnovu izmerenog masenog protoka vazduha pored leptira, prema izrazu

$$\dot{m}_{fi} = \frac{\dot{m}_{at}}{\lambda_{zelj} \cdot L_{th}} \quad (5.44)$$

jer ECU motora „ne zna“ za postojanje još jedne struje vazduha koja ulazi u usisni kolektor. S obzirom na to, da je protok vazduha kroz usisni kolektor veći od onog, kojeg je izmerio protokomer, smeša postaje siromašnija od predviđene. Stvarni koeficijent viška vazduha ovakve smeše je

$$\lambda_{stv} = \frac{(\dot{m}_{at} + \dot{m}_{fv})}{\dot{m}_{fi} \cdot L_{th}} \quad (5.45)$$

gde je  $\dot{m}_{fv}$  maseni protok „falš“-vazduha, koji se u simulatoru neispravnosti određuje iz jednačine (5.28).

## Umanjen EGR protok

Vrednost potrebnog masenog protoka ubrizganog goriva računa se na osnovu masenog protoka vazduha pored leptira, kao u jednačini (5.44). Pri umanjenom EGR protoku  $\dot{m}_{EGR}$  u odnosu na maksimalni  $\dot{m}_{EGRmax}$ , menja se u izvesnoj meri maseni protok vazduha pored leptira  $\dot{m}_{at}$ . Ovo ima uticaja na doziranje goriva, ali tako izmenjena vrednost već ulazi u izraz (5.44), tako da nema potrebe za nekim dodatnim jednačinama, koje bi uzele u obzir umanjeni EGR protok.

### 5.3.7.2 Motor bez protokomera, otvorena sprega sa $\lambda$ -sondom

Kod motora bez protokomera vazduha, potrebni maseni protok ubrizganog goriva  $\dot{m}_{fi}$  određuje se na osnovu izračunatog masenog protoka vazduha kroz usisni kolektor  $\dot{m}_{ap}$  i maksimalnog masenog EGR protoka  $\dot{m}_{EGRmax}$ , prema jednačini [113]

$$\dot{m}_{fi} = \frac{\dot{m}_{ap} - \dot{m}_{EGRmax}}{\lambda_{zelj} \cdot L_{th}} \quad (5.46)$$

Jednačina (5.46) važi za slučaj, kada nema neispravnosti u usisnom sistemu.

Kada motor bez protokomera radi u režimu otvorene sprege sa lambda-sondom, na doziranje goriva utiču sledeće neispravnosti:

- pogrešno očitavanje senzora pritiska u usisnom kolektoru,
- pogrešno očitavanje senzora temperature u usisnom kolektoru,
- ulaz „falš“-vazduha u usisni sistem i
- umanjen EGR protok.

## Pogrešno očitavanje davača pritiska u usisnom kolektoru

Usled pogrešnog očitavanja davača pritiska u usisnom kolektoru, izračunata vrednost potrebnog masenog protoka ubrizganog goriva će biti

$$\dot{m}_{fi} = \frac{(\dot{m}_{ap})_{pGreš} - \dot{m}_{EGRmax}}{\lambda_{zelj} \cdot L_{th}} \quad (5.47)$$

gde je  $(\dot{m}_{ap})_{pGreš}$  maseni protok kroz usisni kolektor, izračunat na osnovu pogrešnog očitavanja davača pritiska, a  $\dot{m}_{EGRmax}$  je maksimalni maseni EGR protok za dati režim rada.

Maseni protok  $(\dot{m}_{ap})_{pGreš}$  se u simulatoru neispravnosti određuje prema jednačini (5.33), a  $\dot{m}_{EGRmax}$  prema jednačini (5.38).

Ostvarivanjem ovako izračunatog potrebnog protoka goriva, dobiće se smeša drugačijeg sastava od željenog. Taj sastav smeše definisan je stvarnim koeficijentom viška vazduha,  $\lambda_{stv}$ . Vrednost  $\lambda_{stv}$  određena je izrazom

$$\lambda_{stv} = \frac{\dot{m}_{ap} - \dot{m}_{EGRmax}}{\dot{m}_{fi} \cdot L_{th}} \quad (5.48)$$

gde je  $\dot{m}_{ap}$  stvarna vrednost masenog protoka kroz usisni kolektor, a ne ona koja je izračunata na osnovu pogrešnog očitavanja davača pritiska.

### Pogrešno očitavanje davača temperature u usisnom kolektoru

Potreban maseni protok ubrizganog goriva, u slučaju pogrešnog očitavanja davača temperature u usisnom kolektoru, određuje se iz jednačine

$$\dot{m}_{fi} = \frac{(\dot{m}_{ap})_{TGreš} - \dot{m}_{EGRmax}}{\lambda_{želj} \cdot L_{th}} \quad (5.49)$$

gde je  $(\dot{m}_{ap})_{TGreš}$  vrednost masenog protoka kroz usisni kolektor, izračunata na osnovu pogrešnog očitavanja temperature. U simulatoru neispravnosti, ova vrednost se dobija iz jednačine (5.35).

Ostvarivanjem masenog protoka goriva izračunatog u jednačini (5.49), dobija se smeša, čiji sastav odstupa od željenog. Tako dobijena smeša definisana je stvarnim koeficijentom viška vazduha

$$\lambda_{stv} = \frac{\dot{m}_{ap} - \dot{m}_{EGRmax}}{\dot{m}_{fi} \cdot L_{th}} \quad (5.50)$$

gde je  $\dot{m}_{ap}$  stvarna vrednost masenog protoka kroz usisni kolektor, a ne ona koja je izračunata na osnovu pogrešnog očitavanja davača temperature.

### Ulaz „falš“-vazduha u usisni sistem

U slučaju pojave ove neispravnosti, postoji značajna razlika između motora sa masenim protokomerom i motora bez protokomera.

Kod motora sa masenim protokomerom, taj uređaj ne registruje „falš“-vazduh, jer je to vazduh koji kroz njega ni ne prolazi. Međutim, „falš“-vazduh učestvuje u formiranju smeše i remeti njen željeni sastav.

Kod motora bez protokomera, o kojima se govori u ovom odeljku, „falš“-vazduh ne ulazi „neopaženo“ u cilindre motora. On će biti registrovan pomoću davača pritiska u usisnom kolektoru, jer će svojim prisustvom uticati na ovu veličinu. Tako će ECU motora moći da prilagodi doziranje goriva novonastaloj situaciji, pa ova neispravnost ne bi trebalo da utiče na sastav smeše. Međutim, svakako će uticati na promenu praćenih veličina u simulatoru neispravnosti.

Kada postoji otvor kroz koji u usisni sistem ulazi neželjeni vazduh, potrebni maseni protok ubrizganog goriva  $\dot{m}_{fi}$  određuje se prema izrazu

$$\dot{m}_{fi} = \frac{(\dot{m}_{ap})_{falš} - \dot{m}_{EGRmax}}{\lambda_{želj} \cdot L_{th}} \quad (5.51)$$

gde je  $(\dot{m}_{ap})_{falš}$  izračunati maseni protok vazduha kroz usisni kolektor, koji u sebi sadrži i maseni protok „falš“-vazduha. S obzirom da se sistem doziranja goriva, na osnovu

jednačine (5.51), prilagodio postojećoj neispravnosti u usisnom sistemu, stvarni koeficijent viška vazduha će u ovom slučaju biti jednak željenoj vrednosti,

$$\lambda_{\text{stv}} = \lambda_{\text{želj}}. \quad (5.52)$$

### Umanjen EGR protok

U ovom slučaju, vrednost potrebnog masenog protoka ubrizganog goriva računa se na osnovu izračunatog masenog protoka kroz usisni kolektor  $\dot{m}_{\text{ap}}$  i maksimalnog masenog EGR protoka,  $\dot{m}_{\text{EGRmax}}$ , za dati režim rada motora, prema jednačini

$$\dot{m}_{\text{fi}} = \frac{\dot{m}_{\text{ap}} - \dot{m}_{\text{EGRmax}}}{\lambda_{\text{želj}} \cdot L_{\text{th}}}. \quad (5.53)$$

Međutim, pošto je EGR protok umanjen u odnosu na maksimalni  $\dot{m}_{\text{EGRmax}}$  i iznosi  $\dot{m}_{\text{EGR}}$ , neće biti ostvaren kvalitet smeše sa željenom vrednošću koeficijenta viška vazduha. Njegova stvarna vrednost će biti

$$\lambda_{\text{stv}} = \frac{\dot{m}_{\text{ap}} - \dot{m}_{\text{EGR}}}{\dot{m}_{\text{fi}} \cdot L_{\text{th}}} \quad (5.54)$$

pri čemu se veličina  $\dot{m}_{\text{EGR}}$  u *NSimulator*-u određuje iz jednačine (5.39).

#### 5.3.7.3 Rad motora u zatvorenoj sprezi sa $\lambda$ -sondom

Osnovna karakteristika rada motora u zatvorenoj sprezi sa lambda-sondom je ta, da se vrši korekcija doziranja goriva, ukoliko stvarna vrednost koeficijenta viška vazduha  $\lambda_{\text{stv}}$  odstupa od željene vrednosti  $\lambda_{\text{želj}}$ . Ova korekcija se u simulatoru neispravnosti vrši korigovanjem veličine potrebnog masenog protoka ubrizganog goriva  $\dot{m}_{\text{fi}}$ , a tako korigovana veličina imaće oznaku  $\dot{m}_{\text{fikor}}$ .

Na ovom mestu bi trebalo raščistiti neke terminološke stvari. Naime, pri radu motora u otvorenoj sprezi sa lambda-sondom, *stvarni koeficijent viška vazduha*,  $\lambda_{\text{stv}}$ , definiše sastav smeše, nastale usled postojanja neispravnosti u usisnom sistemu. Pri radu motora u otvorenoj sprezi, to je i konačna vrednost koeficijenta viška vazduha. Međutim, u slučaju režima rada u zatvorenoj sprezi sa lambda-sondom, to nije njegova konačna vrednost, jer treba da se izvrši korekcija. Zbog toga će vrednost koeficijenta viška vazduha, koja je u režimu otvorene sprege bila označena sa  $\lambda_{\text{stv}}$ , ovde biti označena sa  $\lambda_{\text{nkr}}$  i nosiće naziv *nekorigovani koeficijent viška vazduha*. Znači, važi identitet  $\lambda_{\text{nkr}} = \lambda_{\text{stv}}$ . Vrednost, koju koeficijent viška vazduha dobije posle korekcije, biće nazvana konačnom i imaće oznaku  $\lambda_{\text{kon}}$ .

Kod simulacije neispravnosti u usisnom sistemu, pri radu motora u režimu zatvorene sprege sa lambda-sondom, važe sve jednačine od (5.42)-(5.45) u slučaju motora sa masenim protokomerom i jednačine (5.46)-(5.54) u slučaju motora bez protokomera, uz napomenu da oznaka  $\lambda_{\text{stv}}$  svuda treba da bude zamenjena sa  $\lambda_{\text{nkr}}$ . Ove jednačine koristiće se za izračunavanje veličina  $\dot{m}_{\text{fi}}$  i  $\lambda_{\text{nkr}}$ , u zavisnosti od neispravnosti koja se simulira i u zavisnosti od načina određivanja masenog protoka vazduha kroz usisni sistem motora.

Jednostavniji slučaj korekcije vrednosti  $\lambda$  je kada se sastav smeše može korigovati bez ograničenja. Drugim rečima, moguće je tako korigovati maseni protok ubrizganog goriva, da koeficijent viška vazduha dostigne željenu vrednost  $\lambda_{\text{želj}}$ , bez obzira na veličinu potrebne korekcije. Da bi se došlo do jednačine, na osnovu koje će se vršiti korekcija, treba upariti izraze za  $\lambda_{\text{nkr}}$  i  $\lambda_{\text{želj}}$ :

$$\lambda_{\text{nkr}} = \frac{\dot{m}_{\text{at}}}{\dot{m}_{\text{fi}} \cdot L_{\text{th}}} \quad (5.55)$$



$$\lambda_{\text{želj}} = \frac{\dot{m}_{\text{at}}}{\dot{m}_{\text{fikor}} \cdot L_{\text{th}}} \quad (5.56)$$

iz kojih sledi

$$\frac{\dot{m}_{\text{at}}}{L_{\text{th}}} = \lambda_{\text{nkr}} \cdot \dot{m}_{\text{fi}} = \lambda_{\text{želj}} \cdot \dot{m}_{\text{fikor}} \quad (5.57)$$

i na kraju konačan izraz

$$\dot{m}_{\text{fikor}} = \dot{m}_{\text{fi}} \cdot \frac{\lambda_{\text{nkr}}}{\lambda_{\text{želj}}} \quad (5.58)$$

Postoji i opštiji slučaj u kojem treba izvršiti korekciju doziranja goriva. Naime, neki sistemi za upravljanje doziranjem goriva dozvoljavaju samo ograničenu korekciju, tj. postoji neka maksimalno dozvoljena korekcija koja se može izvesti,  $\Delta\lambda_{\text{doz}}$ , koja ide do 25% [114]. Drugim rečima, ako je razlika između  $\lambda_{\text{nkr}}$  i  $\lambda_{\text{želj}}$  tolika, da je za izjednačavanje te dve vrednosti potrebna korekcija veća od dozvoljene, onda koeficijent viška vazduha neće ni dostići željenu vrednost  $\lambda_{\text{želj}}$ . Tada će se korekcija izvršiti samo do neke limitirane vrednosti.

Algoritam, po kojem se određuje konačna vrednost koeficijenta viška vazduha  $\lambda_{\text{kon}}$ , za slučaj rada motora u zatvorenoj sprezi sa lambda-sondom, dat je na Sl. 5.4, a u nastavku sledi objašnjenje.

Najpre se odredi relativna korekcija  $\Delta\lambda_{\text{rel}}$ , koju treba izvršiti nad  $\lambda_{\text{nkr}}$  da bi se dostigla vrednost  $\lambda_{\text{želj}}$ , prema izrazu

$$\Delta\lambda_{\text{rel}} = \frac{|\lambda_{\text{nkr}} - \lambda_{\text{želj}}|}{\lambda_{\text{nkr}}} \quad (5.59)$$

Zatim se porede vrednosti  $\lambda_{\text{nkr}}$  i  $\lambda_{\text{želj}}$ , kako bi se ustanovilo, da li će korekcija ići u smeru obogaćenja ili osiromašenja. Ako su ove dve vrednosti jednake, onda nema potrebe za korekcijom i tada je praktično  $\lambda_{\text{kon}} = \lambda_{\text{nkr}} = \lambda_{\text{želj}}$ .

Ukoliko vrednosti  $\lambda_{\text{nkr}}$  i  $\lambda_{\text{želj}}$  nisu jednake, treba uporediti potrebnu relativnu korekciju  $\Delta\lambda_{\text{rel}}$  sa maksimalno dozvoljenom korekcijom  $\Delta\lambda_{\text{doz}}$ .

U slučaju da je potrebna korekcija manja od dozvoljene, onda se ona može izvršiti u potpunosti, pri čemu konačna, korigovana vrednost dostiže željenu vrednost koeficijenta viška vazduha,  $\lambda_{\text{kon}} = \lambda_{\text{želj}}$ . Ovo važi i za korekciju u smeru obogaćenja i za korekciju u smeru osiromašenja smeše.

U slučaju da je potrebna korekcija veća od dozvoljene, ona se ne može izvršiti u celosti, što znači da koeficijent viška vazduha ne može dostići vrednost  $\lambda_{\text{želj}}$ . Tada se vrši maksimalno dozvoljena korekcija, kako bi se ostvario sastav smeše što bliži željenom. Konačna vrednost koeficijenta viška vazduha računa se prema jednačini

$$\lambda_{\text{kon}} = \lambda_{\text{nkr}} \cdot (1 + \Delta\lambda_{\text{doz}}), \quad (5.60)$$

kada se korekcija vrši u smeru osiromašenja smeše, a prema jednačini

$$\lambda_{\text{kon}} = \lambda_{\text{nkr}} \cdot (1 - \Delta\lambda_{\text{doz}}), \quad (5.61)$$

kada se korekcija vrši u smeru obogaćenja smeše.

Posle određivanja konačne vrednosti koeficijenta viška vazduha  $\lambda_{\text{kon}}$ , do koje je moguće korigovati smešu, treba izračunati korigovani maseni protok ubrizganog goriva  $\dot{m}_{\text{fikor}}$ , koji je potrebno obezbediti da bi se dobila vrednost  $\lambda_{\text{kon}}$ . Da bi se to moglo uraditi, primeniće se definicija koeficijenta viška vazduha na situaciju pre korekcije,

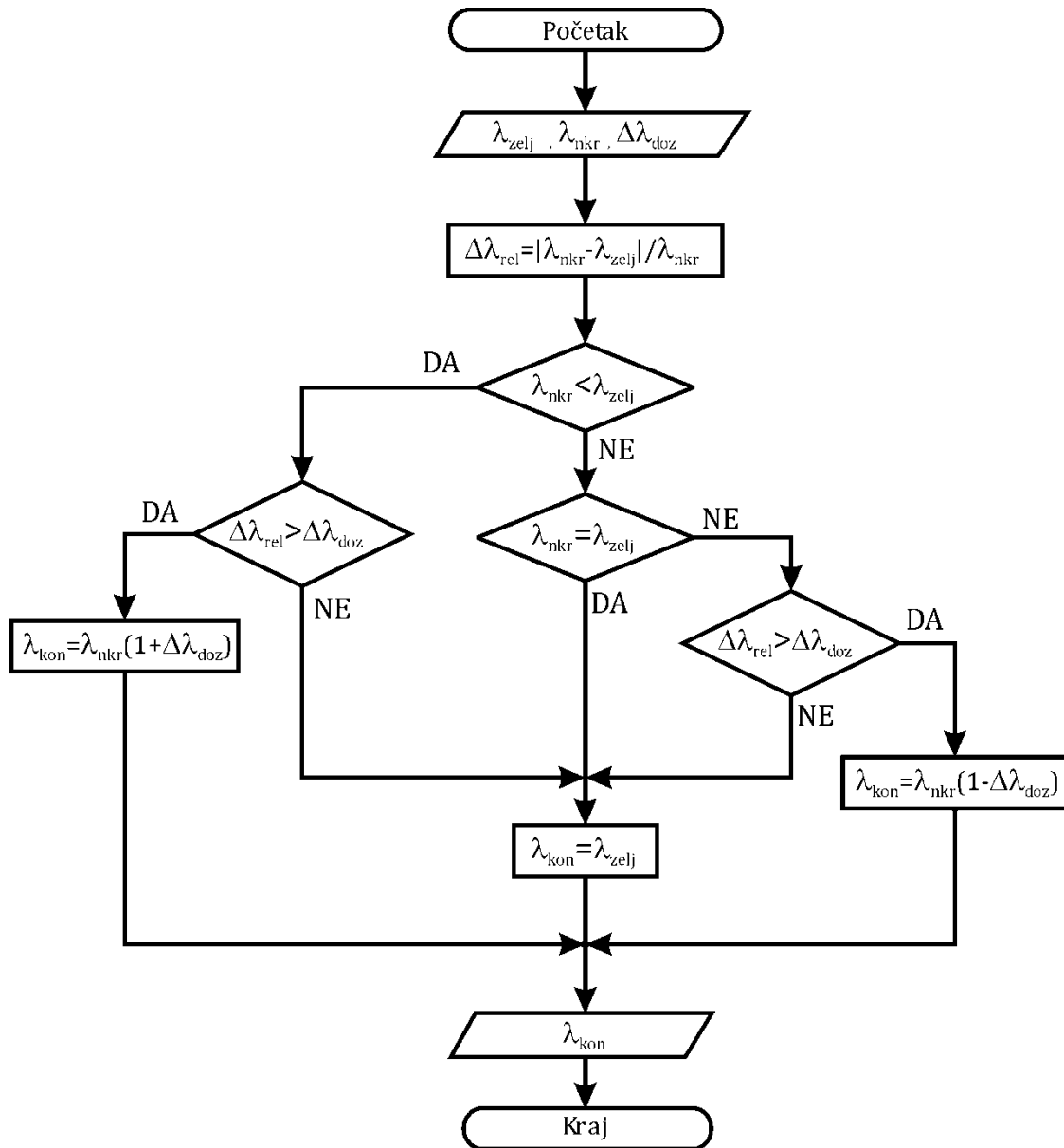
$$\lambda_{\text{nkr}} = \frac{\dot{m}_{\text{at}}}{\dot{m}_{\text{fi}} \cdot L_{\text{th}}} \quad (5.62)$$

i situaciju posle korekcije,

$$\lambda_{kon} = \frac{\dot{m}_{at}}{\dot{m}_{fikor} \cdot L_{th}} \quad (5.63)$$

Iz ove dve jednačine lako se dolazi do izraza za određivanje korigovanog masenog protoka ubrzanog goriva,

$$\dot{m}_{fikor} = \dot{m}_{fi} \cdot \frac{\lambda_{nkr}}{\lambda_{kon}} \quad (5.64)$$



Sl. 5.4 Algoritam određivanja konačne korigovane vrednosti koeficijenta viška vazduha,  $\lambda_{kon}$

Algoritam sa Sl. 5.4 može se smatrati generičkim, pa je kao takav implementiran u **NSimulator**-u. Njime su pokrivena sve opcije korigovanja sastava smeše, pa čak i opcija kada nema korekcije, tj. rad u otvorenoj sprezi sa lambda-sondom. To se obezbeđuje izborom parametra  $\Delta\lambda_{doz}$ . Tako na primer, ako je  $\Delta\lambda_{doz} = 0$ , nema nikakve korekcije i simulira se rad motora u otvorenoj sprezi. Ako se za  $\Delta\lambda_{doz}$  izabere neka relativno velika vrednost (npr. 1000), simulira se rad motora sa korigovanjem smeše bez ograničenja. Ukoliko se želi simulirati rad motora sa ograničenom korekcijom sastava smeše, onda se

parametru  $\Delta\lambda_{\text{doz}}$  dodeljuje vrednost koja odgovara maksimalno dozvoljenoj korekciji (npr.  $\Delta\lambda_{\text{doz}} = 0.2$ , ako je maksimalno dozvoljena korekcija 20%).

Izlaganje u odeljku 5.3 odnosi se na samu srž modela simulatora neispravnosti, koja je implementirana u vidu Simulink modela **NSimulator**. Najvažniji moduli **NSimulator**-a, po kojima se on razlikuje od izvornog Hendriksovog modela, prikazani su u prilogu P1. Tema sledećeg odeljka je primena simulatora neispravnosti u izvođenju virtuelnih eksperimenata sa simuliranjem konkretnih neispravnosti, pa će tamo biti više reči o programu **NSimulatorMenadzer**.

## 5.4 Primena simulatora neispravnosti

Direktna primena simulatora neispravnosti u dijagnostičkom kontekstu obezbeđuje generisanje velikog broja podataka neophodnih za formiranje dijagnostičkog modela. Neophodni podaci dobijaju se izvođenjem velikog broja virtuelnih eksperimenata, pri čemu jedan virtuelni eksperiment predstavlja izvršenje jedne simulacije pomoću modela **NSimulator**. Za izvršenje svake simulacije, neophodni su određeni ulazni podaci, kojih je, u ovom slučaju, popriličan broj. Zbog toga je ulaznim podacima posvećen poseban odeljak, koji sledi u nastavku.

### 5.4.1 Ulazni podaci modela **NSimulator**

Zbog relativno velikog broja ulaznih podataka, oni su, radi lakšeg sagledavanja, sistematizovani u više grupa:

- Konstantni ulazni podaci,
- Parametri za izbor toka simulacije,
- Parametri za izbor neispravnosti koja se simulira i
- Parametri za izbor radne tačke motora.

Konstantni ulazni podaci su ili nasleđeni iz Hendriksovog osrednjenog modela motora ili su novousvojeni, u skladu sa potrebama istraživanja. Vrednosti konstantnih ulaznih podataka koje su korišćene u ovom istraživanju date su u *Tab. 5.1* i one su iste u svim redovnim virtuelnim eksperimentima. Kad se kaže „redovnim“, misli se na virtuelne eksperimente u kojima se generišu podaci za dijagnostički model. Eventualno, kod provere robusnosti dijagnostičkog modela, može doći do variranja nekog od podataka iz *Tab. 5.1*, ali o tome više kada se bude govorilo o testiranju modela.

*Tab. 5.1 Konstantni ulazni podaci korišćeni u virtuelnim eksperimentima*

Naziv	Oznaka	Vrednost	Jedinice
Stehiometrijski odnos vazduha i goriva	$L_{th}$	14.67	-
Eksponent adijabate za vazduh	$\kappa$	1.4	-
Gasna konstanta vazduha	$R$	287	J/(kgK)
Zapremina usisnog kolektora	$V_i$	1.692	dm <sup>3</sup>
Radna zapremina motora	$V_d$	1.275	dm <sup>3</sup>
Pritisak okoline	$p_a$	1	bar
Temperatura recirkulisanih izduvnih gasova	$T_{EGR}$	448	K
Koeficijent maksimalnog EGR protoka	$k_{EGRmax}$	0.2	-
Stvarni moment inercije motora i radne mašine	$I_{stv}$	0.48	kgm <sup>2</sup>
Donja toplotna moć goriva	$H_u$	43000	kJ/kg

Tab 5.1 (Nastavak)

Masa vozila	$m_V$	1000	kg
Statička vrednost koeficijenta otpora kotrljanja	$f_0$	0.1	-
Koeficijent otpora vazduha	$c_W$	0.35	-
Dinamički radijus točka	$r_D$	0.3	m
Prenosni odnos glavnog prenosnika	$i_{GP}$	3.867	-
Prenosni odnos menjača u drugom stepenu prenosa	$i_{M,II}$	2.053	-
Prenosni odnos menjača u trećem stepenu prenosa	$i_{M,III}$	1.32	-
Prenosni odnos menjača u četvrtom stepenu prenosa	$i_{M,IV}$	0.967	-
Prenosni odnos menjača u petom stepenu prenosa	$i_{M,V}$	0.794	-
Koeficijent iskorišćenja transmisije	$\eta_{TR}$	0.9	-
Čeona površina vozila	$A$	2	m <sup>2</sup>
Koeficijent protoka kroz otvor za „falš“-vazduh	$\mu$	0.96	-
Ugao uzdužnog nagiba puta	$\beta$	0	°
Ugao leptira u zatvorenom položaju	$\alpha_0$	8	°
Početni uslovi za inicijalnu simulaciju	Pritisak u usisnom kolektoru	$p_i(0)$	0.23926 bar
	Temperatura u usisnom kolektoru	$T_i(0)$	293 K
	Broj obrtaja motora	$n(0)$	900 min <sup>-1</sup>
Kritičan odnos pritisaka iza i ispred leptira	$p_c$	0.4125	-
Regresioni koeficijenti iz izraza za maseni protok pored leptira (Jednačina (4.8))	$\dot{m}_{at0}$	0	-
	$\dot{m}_{at1}$	5.9403	-
Regresioni koeficijenti iz izraza za maseni protok kroz usisni kolektor (Jednačina (4.18))	$s_i$	0.961	-
	$y_i$	-0.07	-

Parametri za izbor toka simulacije omogućavaju da se izabere jedan od četiri toka odvijanja simulacije pri izvođenju virtuelnog eksperimenta. Tokovi simulacije zavise od toga da li motor ima maseni protokomer ili ne, kao i od načina postizanja željene vrednosti koeficijenta viška vazduha. Za izbor vrste motora, uveden je jedan logički parametar sa izvornim nazivom *imaProtokomer*, a za izbor lambda-režima realni parametar sa izvornim nazivom *deltaLambdaDoz* (u jednačinama (5.60) i (5.61) označen sa  $\Delta\lambda_{doz}$ ). Mogući tokovi simulacije, kao i kombinacija vrednosti parametara koja obezbeđuje realizaciju svakoga od njih, dati su u Tab. 5.2.

Tab. 5.2 Parametri za izbor toka simulacije

Tok simulacije	Kombinacija vrednosti parametara	
Simulacija rada motora bez protokomera vazduha u otvorenoj sprezi sa lambda-sondom	<i>imaProtokomer</i> =0	<i>deltaLambdaDoz</i> =0
Simulacija rada motora bez protokomera vazduha u zatvorenoj sprezi sa lambda-sondom	<i>imaProtokomer</i> =0	<i>deltaLambdaDoz</i> >0
Simulacija rada motora sa masenim protokomerom vazduha u otvorenoj sprezi sa lambda-sondom	<i>imaProtokomer</i> =1	<i>deltaLambdaDoz</i> =0
Simulacija rada motora sa masenim protokomerom vazduha, u zatvorenoj sprezi sa lambda-sondom	<i>imaProtokomer</i> =1	<i>deltaLambdaDoz</i> >0

Svaka vrednost parametra *deltaLambdaDoz* koja je veća od nule, omogućava simulaciju rada motora u zatvorenoj sprezi sa lambda-sondom. Konkretna realna vrednost ovog

parametra pokazuje kolika je maksimalno dozvoljena korekcija vrednosti  $\lambda$ . Ako se želi omogućiti postizanje željene vrednosti  $\lambda_{\text{želj}}$ , onda treba izabrati relativno veliku vrednost  $\text{deltaLambdaDoz}$ . Ovde je to i urađeno, pa je usvojena vrednost  $\text{deltaLambdaDoz}=1000$ . Inače, željene vrednosti koeficijenta viška vazduha u zavisnosti od broja obrtaja i opterećenja motora, smeštene su u mapi koja je ugrađena u *NSimulator*, a preuzeta je iz sistema upravljanja stvarnog motora koji je poslužio za laboratorijski deo istraživanja u ovom radu. Ova mapa je prikazana u *Tab. 5.3*.

*Tab. 5.3 Mapa željenih vrednosti koeficijenta viška vazduha*

$\lambda_{\text{želj}}$		Opterećenje [%]				
		20	40	60	80	100
Broj obrtaja motora	700	1	1	1	0.95	0.9
	1000	1	1	1	0.95	0.9
	2000	1	1	1	0.95	0.9
	3000	1	1	1	0.95	0.9
	4100	1	1	1	0.95	0.9
	4900	1	1	0.96	0.95	0.9
	6100	1	1	0.96	0.96	0.9
	6700	1	1	0.95	0.97	0.9

Treću grupu ulaznih podataka čine parametri za izbor neispravnosti koja se želi simulirati. Svi ti parametri pominju se u jednačinama za simuliranje odgovarajućih neispravnosti u odeljcima 5.3.2 - 5.3.6. Ovde se, zbog preglednosti, na jednom mestu daju njihovi nazivi, oznake, podrazumevane vrednosti, kao i brojevi jednačina u kojima ti parametri figurišu (*Tab. 5.4*). Kad se kaže „podrazumevana vrednost parametra“, misli se na vrednost koju taj parametar ima, kada u usisnom sistemu nema nikakvih neispravnosti.

*Tab. 5.4 Podrazumevane vrednosti parametara za izbor neispravnosti*

Parametar	Oznaka u tekstu	Ime promenljive u modelu	Podrazumevana vrednost	Broj jednačine
Prečnik otvora kroz koji ulazi „falš“-vazduh	$d_{fv}$	$\text{precnikOtvora}$	0	(5.28)
Greška pojačanja senzora pritiska u usisnom kolektoru	$G_{\text{pgain}}$	$k\text{SensPritGain}$	1	(5.33) i (5.34)
Greška ofseta senzora pritiska u usisnom kolektoru	$G_{\text{poffset}}$	$k\text{SensPritOffset}$	0	(5.33) i (5.34)
Greška pojačanja senzora temperature u usisnom kolektoru	$G_{\text{Tgain}}$	$k\text{SensTempGain}$	1	(5.35)
Greška ofseta senzora temperature u usisnom kolektoru	$G_{\text{Toffset}}$	$k\text{SensTempOffset}$	0	(5.35)
Greška pojačanja masenog protokomera vazduha	$G_{\text{MAFgain}}$	$k\text{SensMAFgain}$	1	(5.36)
Greška ofseta masenog protokomera vazduha	$G_{\text{MAFoffset}}$	$k\text{SensMAFoffset}$	0	(5.36)
Koeficijent umanjenog EGR protoka	$k_{\text{EGR}}$	$k_{\text{EGR}}$	1	(5.39)

Dakle, svim parametrima iz *Tab. 5.4* dodeljuju se podrazumevane vrednosti, kada se želi simulirati rad motora bez neispravnosti u usisnom sistemu. Ukoliko se želi simulirati rad motora sa nekom neispravnošću, onda se ključnom parametru za tu neispravnost, dodeljuje odgovarajuća vrednost, koja omogućava simulaciju te neispravnosti. Sve konkretne neispravnosti koje su u ovom istraživanju simulirane u cilju razvoja

dijagnostičkog modela, navedene su, zajedno sa svojim ključnim parametrima, u *Tab. 5.5*. Kao što se može videti iz tabele, u pitanju je pet vrsta neispravnosti, sa po nekoliko različitih intenziteta od svake vrste, što čini ukupno 17 neispravnosti.

*Tab. 5.5 Vrednosti ključnih parametara pri simulacijama konkretnih neispravnosti*

Naziv neispravnosti	Vrednost ključnog parametra
MAP senzor očitava za 10 kPa nižu vrednost	$G_{\text{poffset}} = -10$
MAP senzor očitava za 5 kPa nižu vrednost	$G_{\text{poffset}} = -5$
MAP senzor očitava za 5 kPa višu vrednost	$G_{\text{poffset}} = +5$
MAP senzor očitava za 10 kPa višu vrednost	$G_{\text{poffset}} = +10$
IAT senzor očitava za 15 stepeni nižu vrednost	$G_{\text{Toffset}} = -15$
IAT senzor očitava za 10 stepeni nižu vrednost	$G_{\text{Toffset}} = -10$
IAT senzor očitava za 10 stepeni višu vrednost	$G_{\text{Toffset}} = +10$
IAT senzor očitava za 15 stepeni višu vrednost	$G_{\text{Toffset}} = +15$
MAF senzor očitava 10 % nižu vrednost	$G_{\text{MAFgain}} = 0.9$
MAF senzor očitava 5 % nižu vrednost	$G_{\text{MAFgain}} = 0.95$
MAF senzor očitava 5 % višu vrednost	$G_{\text{MAFgain}} = 1.05$
MAF senzor očitava 10 % višu vrednost	$G_{\text{MAFgain}} = 1.1$
Ulaz „falš“-vazduha kroz otvor prečnika 2mm	$d_{\text{fv}} = 0.002$
Ulaz „falš“-vazduha kroz otvor prečnika 4mm	$d_{\text{fv}} = 0.004$
Ulaz „falš“-vazduha kroz otvor prečnika 6mm	$d_{\text{fv}} = 0.006$
EGR protok umanjen na 80%	$k_{\text{EGR}} = 0.8$
EGR protok umanjen na 60%	$k_{\text{EGR}} = 0.6$

Kada se želi simulirati neka od navedenih neispravnosti, samo ključnom parametru za simulaciju te neispravnosti dodeljuje se odgovarajuća vrednost iz *Tab. 5.5*, a svim ostalim parametrima za izbor neispravnosti dodeljuju se podrazumevane vrednosti iz *Tab. 5.4*.

Osim parametara za izbor toka simulacije i parametara za izbor neispravnosti koja se želi simulirati, tu su još i parametri za izbor radne tačke motora. Naime, jedna od pretpostavki pri razvoju simulatora neispravnosti (Odeljak 5.1) bila je i ta, da motor radi na ustaljenom režimu. Otuda će se i simulacije neispravnosti izvoditi na ustaljenim režimima rada. Zbog toga je neophodno, pomoću odgovarajućih parametara definisati radnu tačku motora u kojoj će on ustaljeno da radi. Izbor režima rada motora na vozilu vrši se direktno pomoću otvora leptira i stepena prenosa. Ova dva parametra imaju dominantan uticaj na tri od četiri izlazne veličine simulatora neispravnosti:

- broj obrtaja motora  $n$ ,
- pritisak u usisnom kolektoru  $p_i$  i
- maseni protok vazduha  $\dot{m}_{\text{at}}$ .

Međutim, preostala, četvrta izlazna veličina, temperatura u usisnom kolektoru  $T_i$ , nije pod značajnim uticajem otvora leptira i stepena prenosa, ako važe ranije usvojene pretpostavke da se pri svakoj simulaciji smatra da motor radi na normalnoj radnoj temperaturi i da je maseni EGR protok proporcionalan ukupnom masenom protoku kroz usisni kolektor. Sa aspekta dijagnostike, bilo bi poželjno da se parametrima za izbor radne tačke motora pridruži još jedan, koji ima uticaja na temperaturu u usisnom kolektoru. Temperatura okoline,  $T_a$ , je zadovoljavajući izbor, jer se spoljašnji vazduh temperature  $T_a$  u usisnom sistemu meša sa recirkulisanim izduvnim gasovima temperature  $T_{\text{EGR}}$  i svakako učestvuje u

formiranju temperaturnog stanja u usisnom kolektoru. Osim toga, ni uticaj temperature okoline na gustinu vazduha nije zanemariv, kada se posmatra raspon temperatura koje se dostižu u toku godine. Zbog svega navedenog, ovde će u parametre za izbor radne tačke motora biti uvršteni:

- otvor leptira (u procentima),  $\alpha_{proc}$ ,
- stepen prenosa,  $SP$  i
- temperatura okoline,  $T_a$ .

Temperatura okoline u realnoj eksploataciji vozila se ne može birati (za razliku od otvora leptira i stepena prenosa), ali se u eksperimentima simulacija neispravnosti može usvojiti bilo koja vrednost za  $T_a$ , što je dodatno opravdanje za svrstavanje ove veličine u parametre za izbor radne tačke.

O ukupnom broju radnih tačaka motora u kojima će se vršiti simulacije, kao i o konkretnim vrednostima parametara za izbor radne tačke motora, biće govora u poglavlju o dijagnostičkom modelu (Poglavlje 6). Razlog je taj, što i broj radnih tačaka, a i njihov položaj u radnom dijapazonu motora, značajno utiču na mogućnosti za formiranje tog modela, kao i na njegove performanse.

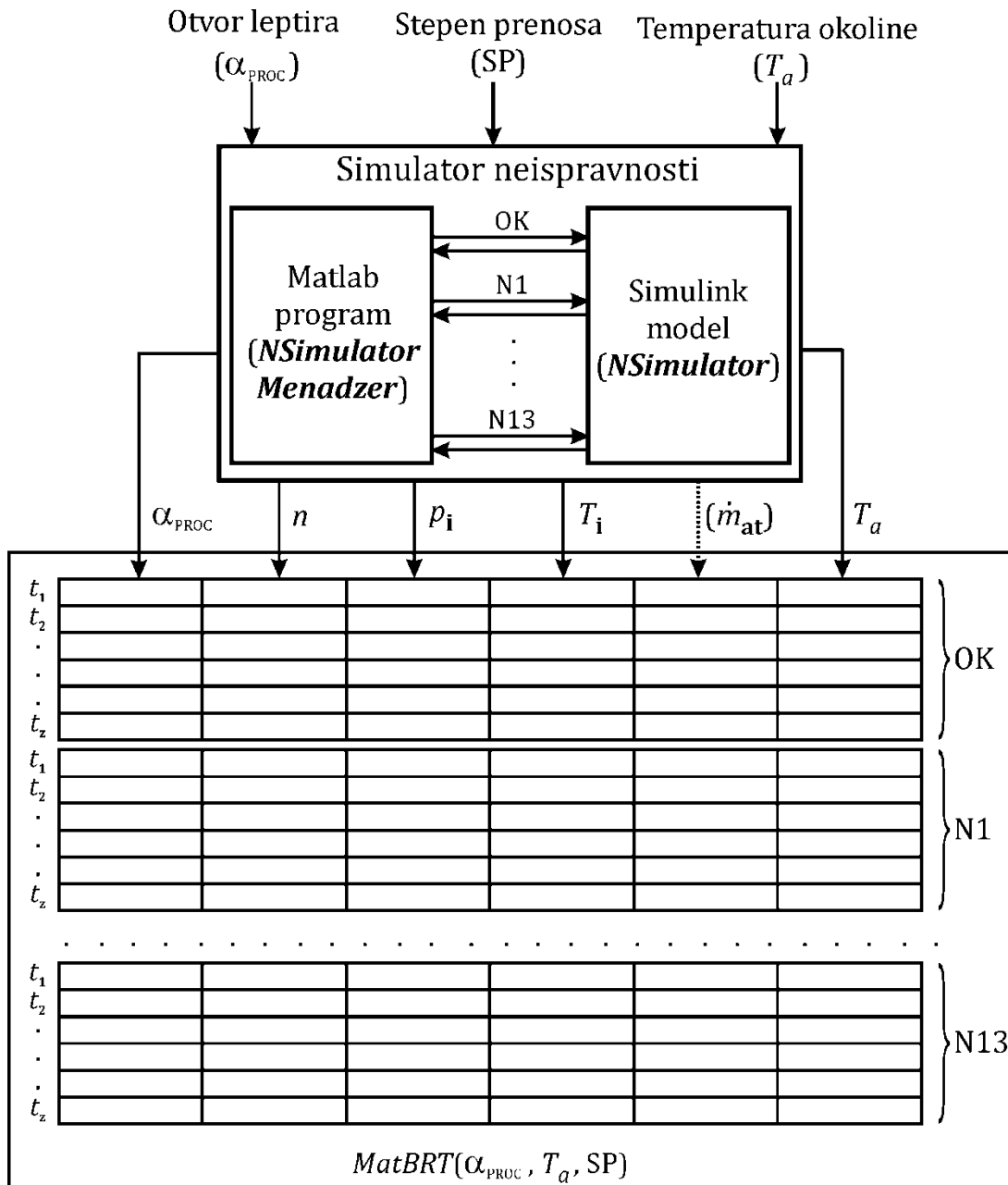
### 5.4.2 Struktura matrice izlaznih podataka simulatora neispravnosti

Svaka pojedinačna simulacija izvodi se jednim pokretanjem Simulink modela *NSimulator*, pri čemu se simulira ili rad motora bez neispravnosti ili rad motora sa nekom neispravnošću u usisnom sistemu. U *Tab. 5.5* navedeno je ukupno 17 neispravnosti koje su uzete u obzir u ovom istraživanju. Međutim, za odabranu vrstu motora (sa masenim protokomerom ili bez) nisu interesantne sve neispravnosti iz *Tab. 5.5*. Zato su u *Tab. 5.6*, za obe vrste motora, navedena stanja usisnog sistema koja će se simulirati za konkretnu odabranu vrstu motora. To iznosi ukupno 14 stanja - jedno bez neispravnosti i još 13 stanja koja karakteriše prisustvo neke od navedenih neispravnosti. U svakoj radnoj tački motora potrebno je simulirati ovih 14 stanja, što znači da u svakoj radnoj tački treba 14 puta pokrenuti model *NSimulator*.

*Tab. 5.6 Neispravnosti u zavisnosti od načina određivanja masenog protoka vazduha*

Stanje usisnog sistema	Neispravnosti kod motora sa masenim protokomerom vazduha	Neispravnosti kod motora bez protokomera vazduha
OK	Nema neispravnosti (Sve OK)	
N1	MAP senzor očitava za 10 kPa nižu vrednost	
N2	MAP senzor očitava za 5 kPa nižu vrednost	
N3	MAP senzor očitava za 5 kPa višu vrednost	
N4	MAP senzor očitava za 10 kPa višu vrednost	
N5	MAF senzor očitava 10 % nižu vrednost	IAT senzor očitava za 15° nižu vrednost
N6	MAF senzor očitava 5 % nižu vrednost	IAT senzor očitava za 10° nižu vrednost
N7	MAF senzor očitava 5 % višu vrednost	IAT senzor očitava za 10° višu vrednost
N8	MAF senzor očitava 10 % višu vrednost	IAT senzor očitava za 15° višu vrednost
N9	Ulaz „falš“-vazduha kroz otvor prečnika 2mm	
N10	Ulaz „falš“-vazduha kroz otvor prečnika 4mm	
N11	Ulaz „falš“-vazduha kroz otvor prečnika 6mm	
N12	EGR protok umanjen na 80%	
N13	EGR protok umanjen na 60%	

Ranije je rečeno da *N Simulator*-om upravlja Matlab program *N Simulator Menadžer*. Ovaj program je koncipiran tako, da se pri jednom njegovom izvršenju simulira sukcesivno svih 14 stanja usisnog sistema u jednoj baznoj radnoj tački motora. Pojam *bazne radne tačke* je uveden, da bi se otklonila eventualna zabuna u daljem tekstu. Pod baznom radnom tačkom ovde se podrazumeva željena kombinacija otvora leptira  $\alpha_{PROC}$ , stepena prenosa  $SP$  i temperature okoline  $T_a$ . Jednoj baznoj radnoj tački odgovara više realnih radnih tačaka u kojima motor može da radi usled prisustva različitih neispravnosti, ali za sve te realne radne tačke zajednički je skup konkretnih vrednosti  $(\alpha_{PROC}, SP, T_a)$ .



Sl. 5.5 Generisanje izlazne matrice simulatora neispravnosti

Šema na Sl. 5.5 ilustruje situaciju koja se odnosi na jedno izvršenje programa *N Simulator Menadžer*. Tokom tog jednog izvršenja, *N Simulator Menadžer* pokreće Simulink model *N Simulator* 14 puta: jednom da bi se simuliralo ispravno funkcionisanje motora i još 13 puta da bi se simulirala svaka neispravnost. Svih 14 simulacija imaju podjednake vremenske periode trajanja  $t_z$  i u okviru svake od njih prikupi se po  $z$  simulacionih uzoraka. Svaki simulacioni uzorak čini šestorka  $(\alpha_{PROC}, n, p_i, T_i, \dot{m}_{at}, T_a)$ , sa



konkretnim vrednostima tih promenljivih u nekom vremenskom trenutku tokom izvršenja simulacije. Ovde treba napomenuti da, ukoliko se radi o simuliranju neispravnosti kod motora koji nema protokomer vazduha, onda šestorka postaje petorka  $(\alpha_{\text{PROC}}, n, p_i, T_i, T_a)$ . Zbog toga je strelica koja obeležava  $\dot{m}_{\text{at}}$  na izlazu iz simulatora ucrtana kao isprekidana.

Na kraju, posle svih 14 simulacija, kao izlaz iz programa *NSimulatorMenadzer* dobija se matrica podataka koji se odnose na jednu baznu radnu tačku,  $MatBRT(\alpha_{\text{PROC}}, T_a, SP)$ , sa 5 ili 6 kolona i  $(14 \cdot z)$  vrsta, kako je prikazano na Sl. 5.5. Pada u oči da su u kolone te matrice uvrštene i ulazne i izlazne veličine simulatora neispravnosti, što je veoma bitno, s obzirom da ovi podaci treba da budu upotrebljeni za obučavanje neuronskih mreža. To omogućava neuronskoj mreži da „sazna“ koje vrednosti izlaznih veličina simulatora odgovaraju kojim vrednostima ulaznih veličina, pri radu motora u određenoj baznoj radnoj tački za svako od 14 posmatranih stanja. Ulazna veličina  $SP$  nije uvrštena među kolone matrice  $MatBRT$ , jer u okviru jedne neuronske mreže stepen prenosa uvek ima istu vrednost, što će kasnije biti pokazano.

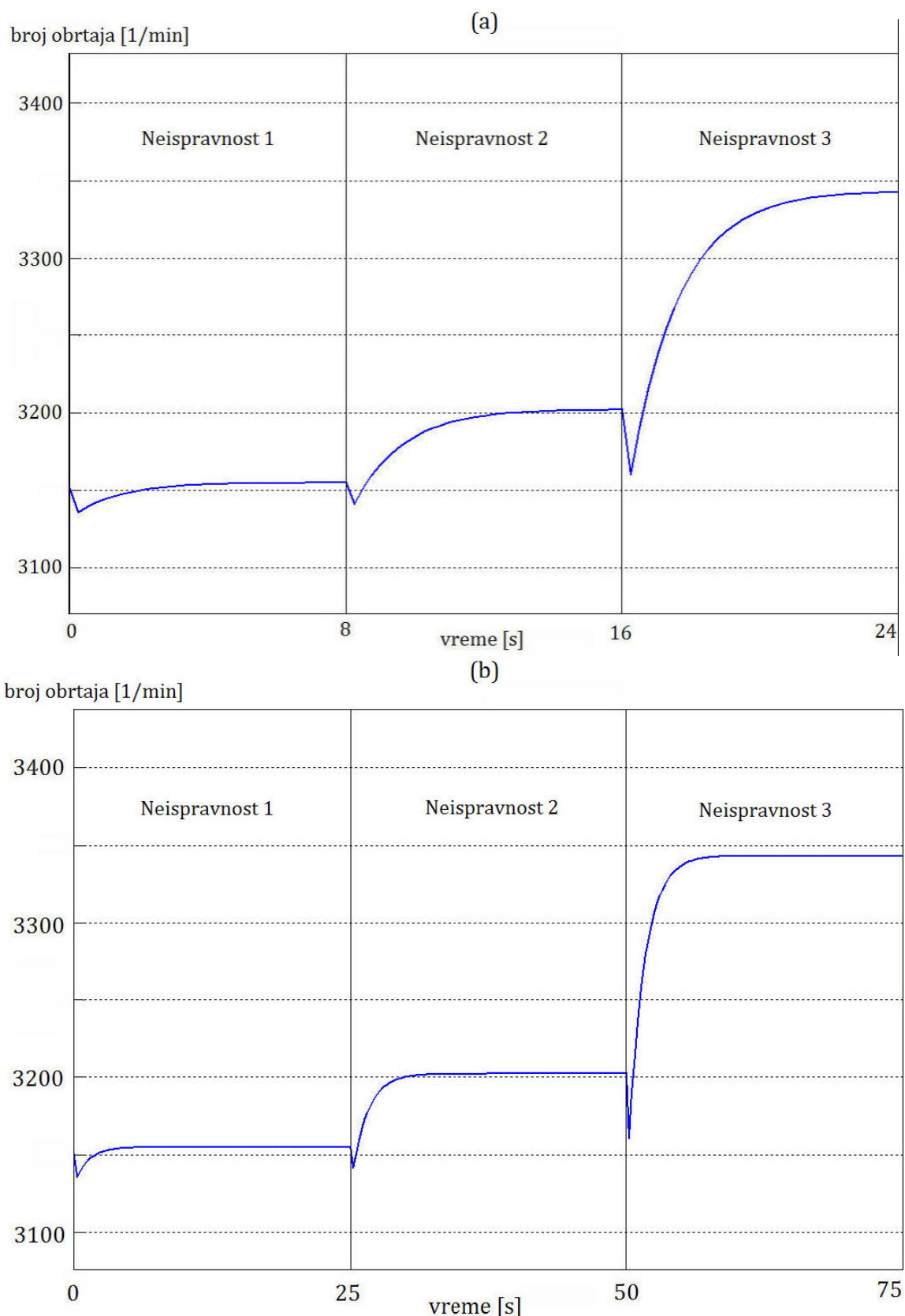
### 5.4.3 Početni uslovi za simulaciju i veličina matrice izlaznih podataka

Pre nego što se malo detaljnije predstavi program *NSimulatorMenadzer*, trebalo bi razjasniti neka pitanja u vezi sa izvršenjem svake pojedinačne simulacije u okviru tog programa. Prvo pitanje se tiče početnih uslova, a drugo je vezano za količinu izlaznih podataka iz simulatora neispravnosti.

Pošto se tokom izvršenja *NSimulator*-a rešavaju diferencijalne jednačine po  $p_i$  (5.31), po  $T_i$  (5.15) i po  $n$  (4.20), potrebni su i početni uslovi za ove veličine. Početni uslovi su nasleđeni iz Hendriksovog osrednjenog modela motora i dati su u Tab. 5.1. Pre zvanične simulacije rada motora u nekoj baznoj radnoj tački, izvršava se jedna inicijalna simulacija rada motora bez neispravnosti u toj istoj radnoj tački, sa početnim uslovima iz Tab. 5.1, u trajanju od 7 sekundi. Kad se kaže *zvanična simulacija*, ovde se misli na simulaciju, čiji će rezultati biti korišćeni u dijagnostičkom modelu. Krajnji uslovi po izvršenju inicijalne simulacije predstavljaju početne uslove za svaku zvaničnu simulaciju u toj baznoj radnoj tački motora. Inicijalna simulacija se izvodi, kako bi se početni uslovi za zvaničnu simulaciju doveli do vrednosti, koje će obezbediti što brže uspostavljanje stanja ustaljenog rada motora u zvaničnoj simulaciji.

Na količinu izlaznih podataka simulatora neispravnosti odlučujući uticaj imaju vreme trajanja simulacije i korak simulacije, koji, u stvari, predstavlja interval uzorkovanja (eng. *sampling time*).

Vreme simulacije je uslovljeno time, da u toku nekog dovoljno dugog perioda, sve izlazne veličine modela *NSimulator* moraju da imaju ustaljene vrednosti. Taj uslov proističe iz pretpostavke da se u ovom istraživanju razvija dijagnostički model koji se odnosi na ustaljeni rad motora. Kod ovih simulacija je problem, što zbog rešavanja diferencijalnih jednačina sa početnim uslovima, ustaljene vrednosti izlaznih veličina *NSimulator*-a ne mogu trenutno da se dostignu. Na početku odvijanja simulacije, izlazne veličine imaju svoje usvojene početne vrednosti, a kako simulacija odmiče, te vrednosti se menjaju, sve dok ne dostignu neke ustaljene vrednosti u skladu sa zadatim parametrima simulacije. Kada se zvanična simulacija izvodi za slučaj rada motora bez neispravnosti, ustaljenost izlaznih veličina se postiže relativno brzo (za manje od jedne sekunde), jer se inicijalna simulacija izvodi upravo za taj slučaj. Međutim, kada se simuliraju neispravnosti, onda je potrebno znatno više vremena da se izlazne veličine *NSimulator*-a ustale, jer se menjaju uslovi u odnosu na inicijalnu simulaciju. Na Sl. 5.6 data su dva primera, gde su u oba simulirane iste tri neispravnosti, samo što je u slučaju pod (a) svaka simulacija trajala po 8 sekundi, a u slučaju pod (b), po 25 sekundi.



Sl. 5.6 Ilustracija dostizanja ustaljenog broja obrtaja pri simulacijama tri različite neispravnosti: (a) kad simulacije traju po 8 sekundi; (b) kad simulacije traju po 25 sekundi

Prvi dijagram pokazuje da je 8 sekundi prekratko vreme za simulaciju, jer se u slučaju sve tri neispravnosti, broj obrtaja povećava tokom simulacije i ne dolazi do uspostavljanja ustaljenog broja obrtaja u toku tog vremenskog perioda. Na drugom dijagramu zapaža se

da posle nekog vremena broj obrtaja dostiže ustaljenu vrednost, koju zadržava do završetka simulacije, kod sve tri neispravnosti. Tokom istraživanja isprobane su još neke vrednosti vremena trajanja simulacije (pored dve malopre pomenute), ali je odlučeno da vreme simulacije bude 25 sekundi, kako bi se osiguralo dobijanje valjanih rezultata simulacije i u slučaju nekih većih neispravnosti.

Na *Sl. 5.6* dati su primeri uspostavljanja ustaljene vrednosti broja obrtaja tokom simulacije, a situacija je slična i sa dostizanjem ustaljenih vrednosti ostalih izlaznih veličina simulatora neispravnosti.

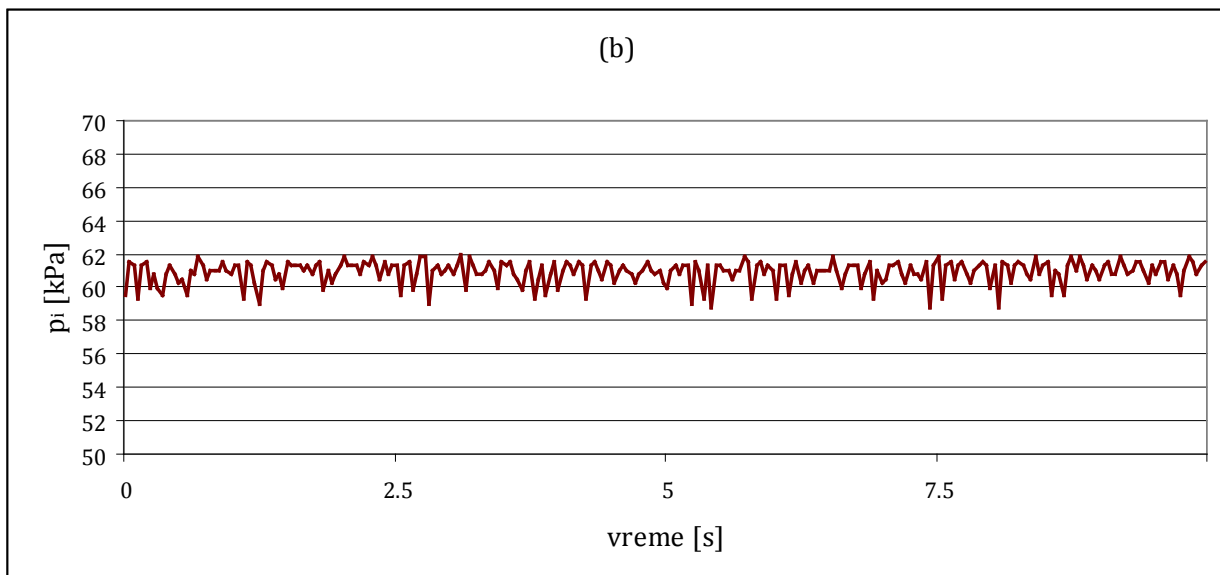
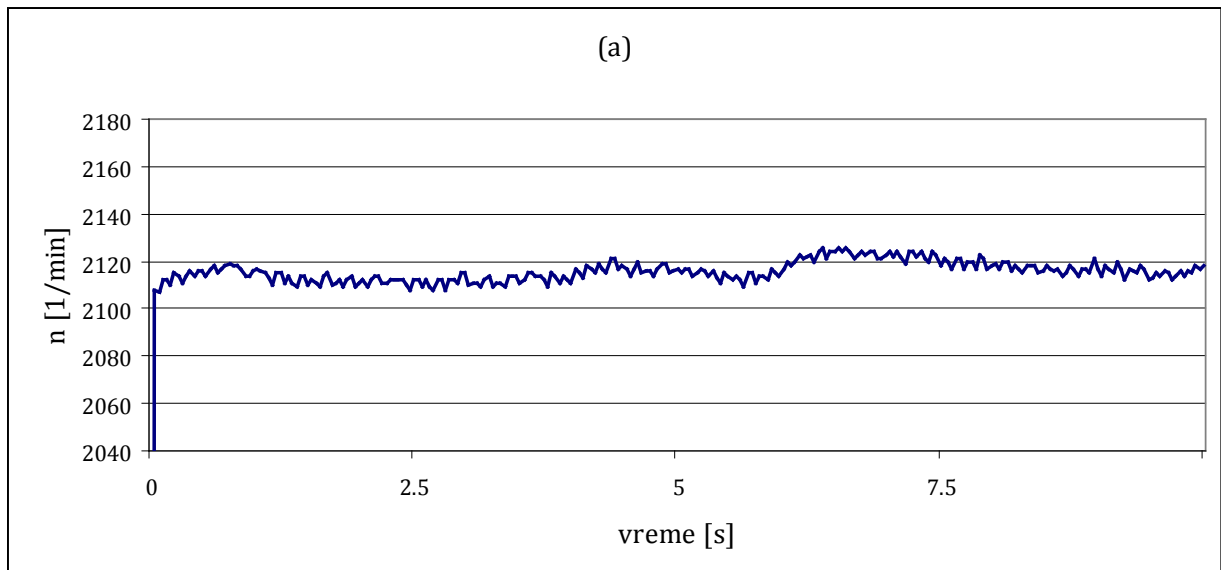
Korak simulacije uslovljen je ukupnom količinom podataka koja treba da posluži za obuku dijagnostičkog modela. U ovom istraživanju, količina podataka je ograničena mogućnostima Matlab-ovog alata za rad sa neuronskim mrežama. Isprobavanjem, posle nekoliko iteracija, ustanovljeno je da broj uzoraka u jednoj simulaciji može da bude najviše 13. Stoga je odlučeno da simulacija obuhvati 3 sekunde rada motora na ustaljenom režimu, sa intervalom uzorkovanja od 0.25 sekundi.

Na prvi pogled, moglo bi se reći da je vreme trajanja simulacije od 3 sekunde kratko, a interval uzorkovanja od 0.25 sekundi predugačak, ako se ima na umu da izlazne veličine simulatora neispravnosti treba da se koriste za obuku neuronskih mreža. To bi zaista bilo tako, kad bi se radilo o prelaznim režimima rada motora. Međutim, kod ustaljenog režima rada, kakav je prikazan na slici *Sl. 5.6b* dovoljan bi bio i samo jedan uzorak, jer je na toj slici prikazana idealizovana ustaljena vrednost broja obrtaja, bez ikakvih pulsacija tokom vremena, što nije realno pri radu stvarnog motora. Rad stvarnog motora karakteriše nepravilan oscilatorni karakter promena u vremenu broja obrtaja motora, pritiska u usisnom kolektoru i masenog protoka vazduha, čak i pri najustaljenijem režimu rada. Kao ilustracija veličine pulsacija, na *Sl. 5.7* su date promene izmerenih vrednosti broja obrtaja i pritiska u usisnom kolektoru motora koji je korišćen za eksperimentalni deo istraživanja, tokom 10 sekundi rada na jednom ustaljenom režimu. Oscilatorni karakter promene vrednosti masenog protoka vazduha pored leptira nije prikazan na *Sl. 5.7*, jer korišćeni motor nema maseni protokomer vazduha. Međutim, kao ilustracija oscilatornog karaktera ove veličine poslužiće *Sl. 5.8*. Na njoj je prikazana fotografija ekrana dijagnostičkog softvera na kojem se prati promena signala masenog protokomera vazduha, izraženog u hercima, tokom 5 sekundi rada motora na vozilu Chevrolet Blazer na ustaljenom režimu [115].

U ovom istraživanju namera je da se simuliraju neispravnosti sa uzimanjem u obzir pulsacija ključnih veličina, poput onih prikazanih na *Sl. 5.7* i *Sl. 5.8*, tako da nikako ne bi bio dovoljan samo jedan uzorak, kao što je slučaj sa idealno ustaljenim radom motora. Međutim, broj od 13 uzoraka po jednoj simulaciji može se smatrati dovoljnim za ovaj slučaj, što će biti pokazano kasnije, pri testiranju dijagnostičkog modela.

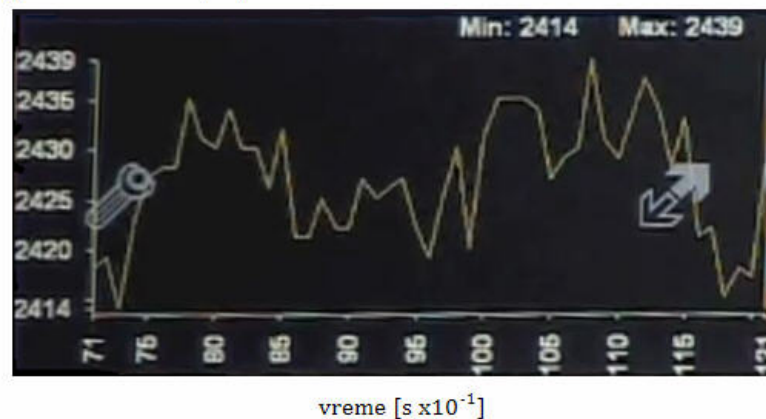
S obzirom da je ranije usvojeno vreme simulacije u trajanju od 25 sekundi, za razvoj dijagnostičkog modela izdvojene su poslednje 3 sekunde ovog perioda, jer se time obezbeđuju ustaljene vrednosti izlaznih veličina *NSimulator*-a, čak i pri simulacijama većih neispravnosti. Dakle, interval od interesa za dijagnostički model počinje na kraju 22. sekunde pa traje do kraja 25. sekunde simulacije.

Kada je utvrđen broj uzoraka po jednoj simulaciji ( $z = 13$ ) ponovnim vraćanjem na *Sl. 5.5*, dolazi se do zaključka da ukupan broj vrsta matrice *MatBRT* iznosi 182 (14 stanja x 13 uzoraka po stanju) i jednak je ukupnom broju uzoraka prikupljenih pri simuliranju rada motora u jednoj baznoj radnoj tački.



Sl. 5.7 Ilustracija promene karakterističnih veličina tokom ustaljenog rada motora: (a) broj obrtaja motora; (b) pritisak u usisnom kolektoru

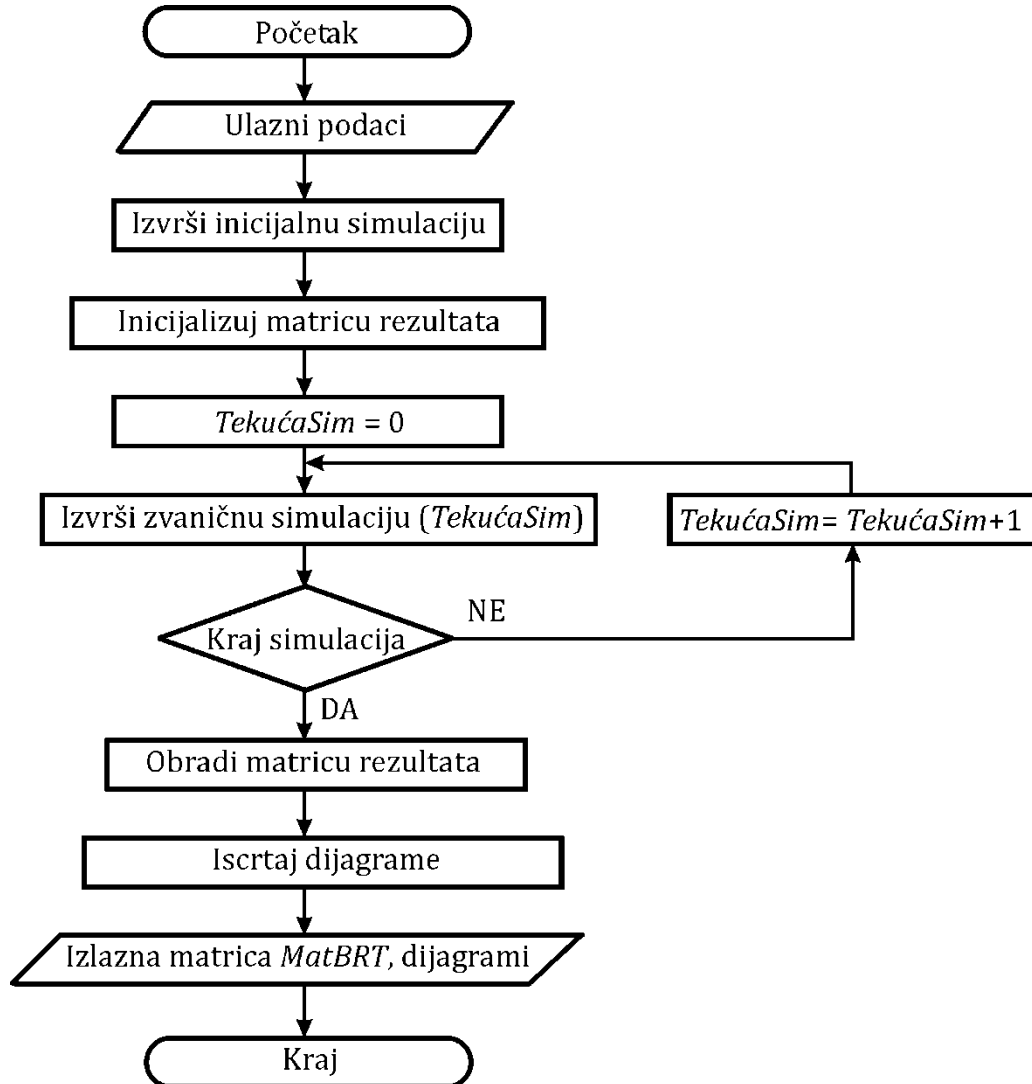
Signal MAF senzora [Hz]



Sl. 5.8 Primer izlaznog signala masenog protokomera vazduha na ustaljenom režimu rada motora [115]

#### 5.4.4 Algoritam programa *NSimulatorMenadzer*

Struktura programa *NSimulatorMenadzer* na globalnom nivou prikazana je na Sl. 5.9, kao kombinacija algoritma i neke vrste pseudokoda. U nastavku je u vidu koraka opisana logika izvršavanja programa sa nekim pojašnjenjima vezanim za svaki korak. Kompletan listing izvornog koda *NSimulatorMenadzer*-a dat je u prilogu P2.



Sl. 5.9 Logika izvršavanja programa *NSimulatorMenadzer*

##### **Korak I: Učitavanje ulaznih podataka**

Najpre se učitavaju konstantni podaci (Tab. 5.1), koji su smešteni u posebnom Matlab fajlu *podaciOMotoru.m*. Ostali ulazni podaci su parametri koji su sadržani u samom *NSimulatorMenadzer*-u, gde se njihove vrednosti mogu menjati, u zavisnosti od toga šta se sve želi simulirati. To su parametri pomoću kojih se definišu:

- način određivanja masenog protoka vazduha (videti Tab. 5.2),
- vrsta lambda režima tokom simulacije (videti Tab. 5.2),
- bazna radna tačka motora,
- vrste i intenziteti neispravnosti koje se simuliraju (videti Tab. 5.4 i Tab. 5.5),
- vreme trajanja i korak simulacije,
- nivo pulsacija broja obrtaja motora, pritiska u usisnom kolektoru i masenog protoka vazduha

O vremenu trajanja i koraku simulacije bilo je reči u prethodnom odeljku, a o parametrima za definisanje nivoa pulsacija izlaznih veličina simulatora neispravnosti biće reči u ovom odeljku malo kasnije.

### **Korak II: Izvršavanje inicijalne simulacije**

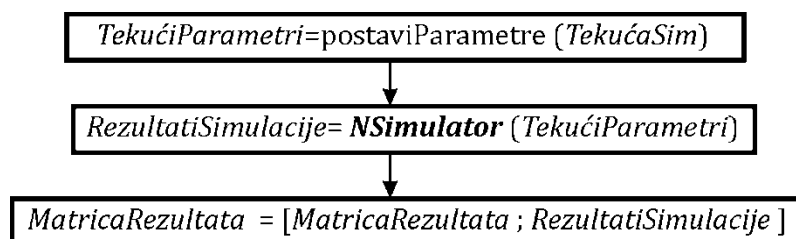
U ovom koraku se izvršava inicijalna simulacija, kako bi se obezbedili početni uslovi za sve druge simulacije u odabranoj baznoj radnoj tački. Inicijalna simulacija izvršava se pozivanjem Simulink modela **NSimulator** za rad motora bez ikakvih neispravnosti i sa sledećim početnim uslovima:  $p_i(0) = 23.926 \text{ kPa}$ ,  $n(0) = 900 \text{ min}^{-1}$ ,  $T_i(0) = T_a$ . Početni uslov  $T_i(0) = T_a$  označava da početna temperatura za izvođenje inicijalne simulacije zavisi od bazne radne tačke motora na koju se simulacija odnosi. Vrednosti koje veličine  $p_i$ ,  $T_i$  i  $n$  imaju na kraju inicijalne simulacije čuvaju se sve vreme tokom izvršenja programa **NSimulatorMenadzer** u odgovarajućoj matrici, odakle se preuzimaju pre izvršenja svake zvanične simulacije.

### **Korak III: Inicijalizacija matrice rezultata i brojača simulacija**

Matricom rezultata nazvana je jedna kumulativna matrica u koju će se sakupljati rezultati zvaničnih simulacija pojedinih stanja. Ona se inicijalizuje tako što postaje prazna matrica, pre izvršenja prve zvanične simulacije,  $MatricaRezultata = [ ]$ . Istovremeno, brojač simulacija, koji je ovde nazvan *TekućaSim* inicijalizuje se postavljanjem na nulu. Inače, vrednosti parametra *TekućaSim* usklađene su sa redosledom stanja neispravnosti iz *Tab. 5.6*, što znači da će se prvo simulirati stanje bez neispravnosti.

### **Korak IV: Izvršenje zvanične simulacije**

Ovaj korak je složeniji od prethodnih, pa će uz objašnjenje biti priložen mali dijagram toka (*Sl. 5.10*) koji se odnosi na taj korak.



*Sl. 5.10 Koraci zvanične simulacije*

Suštinski deo zvanične simulacije čini izvršenje Simulink modela **NSimulator**, ali pre pokretanja ovog modela, neophodno je postaviti relevantne parametre na vrednosti koje odgovaraju tekućoj simulaciji. Te vrednosti parametara nazvane su na *Sl. 5.10* *tekućim parametrima*. Postavljanje parametara podrazumeva:

- dodeljivanje odgovarajućih vrednosti iz *Tab. 5.5* ključnim parametrima za tekuću simulaciju i
- dodeljivanje podrazumevanih vrednosti iz *Tab. 5.4* parametrima, koji nisu relevantni za tekuću simulaciju.

Kada su svi tekući parametri dobili željene vrednosti, poziva se **NSimulator**, koji se izvršava koristeći te vrednosti tekućih parametara. Kao rezultat **NSimulator**-a dobija se matrica, koja je nazvana *RezultatiSimulacije*. Ova matrica ima 13 vrsta, što odgovara ranije usvojenom broju uzoraka po jednoj simulaciji, i 3 ili 4 obavezne kolone, u zavisnosti od načina određivanja masenog protoka vazduha. U obaveznim kolonama su vrednosti veličina  $n$ ,  $p_i$ ,  $T_i$  i eventualno  $\dot{m}_{at}$ , ako se maseni protok vazduha određuje direktnim merenjem (*Sl. 5.5*). Međutim, mogu se matrici *RezultatiSimulacije* dodati još neke opcione kolone, koje će biti interesantne za prikaz rezultata, kao što su maseni protok „falš“-

vazduha, maseni EGR protok, pogrešno očitane vrednosti pritiska, temperature i masenog protoka vazduha pored leptira. Ipak, za razvoj dijagnostičkog modela suštinski su bitne one obavezne kolone.

Na kraju zvanične simulacije jednog stanja, matrica *RezultatiSimulacije* pridružuje se matrici *MatricaRezultata*, tj. vrši se vertikalna konkatenacija ove dve matrice, a rezultat konkatenacije postaje nova *MatricaRezultata* (Sl. 5.10).

### **Korak V: Provera uslova za završetak simulacija**

Ovde se proverava da li je potrebno simulirati još neko stanje. Kad je u pitanju skup stanja dat u Tab. 5.6, konkretan uslov koji se proverava je da li je  $TekućaSim > 13$ . Ako nije, parametar *TekućaSim* se povećava za jedan, kako bi se omogućilo izvršenje simulacije sledećeg stanja, a onda se tok programa vraća na korak IV. Ukoliko je ispunjen uslov  $TekućaSim > 13$ , tok programa se usmerava na korak VI.

### **Korak VI: Obrada matrice rezultata**

Obrada matrice *MatricaRezultata* podrazumeva dodavanje još nekih kolona toj matrici, ali i unošenje efekta pulsacija vrednosti u nekim njenim kolonama.

Kolone koje se dodaju sadrže vrednosti parametara za izbor bazne radne tačke motora:

- Otvor leptira  $\alpha_{PROC}$  i
- Temperatura okoline  $T_a$

Ove dve kolone imaju konstantne vrednosti, jer karakterišu jednu baznu radnu tačku, koja se ne menja tokom jednog izvršavanja programa *NSimulatorMenadzer*. Ove dve kolone su takođe obavezne, da bi odgovarajuća neuronska mreža imala informaciju kojoj baznoj radnoj tački pripada svaki simulacioni uzorak, kako bi se mogla izvršiti interpolacija u slučaju rada motora u nekim nepoznatim radnim tačkama. Pored parametara  $\alpha_{PROC}$  i  $T_a$ , stepen prenosa (*SP*) takođe određuje baznu radnu tačku, ali ga nije potrebno dodavati u vidu posebne kolone. Razlog je relativno mali broj stepeni prenosa u kojima vozilo može da se kreće (u ovom istraživanju nisu uzeti u obzir kontinualni menjački prenosnici), pa će svakom stepenu prenosa u dijagnostičkom modelu odgovarati po jedna neuronska mreža. Otuda, nema potrebe za bilo kakvom interpolacijom po pitanju stepena prenosa, a time ni za dodatnom kolonom u matrici *MatricaRezultata*. Ovo će biti jasnije, kada budu prikazani detalji dijagnostičkog modela (Poglavlje 6).

Što se tiče efekta pulsacija, neophodno ga je uvesti u dijagnostički model, jer promene u vremenu broja obrtaja motora, pritiska u usisnom kolektoru i masenog protoka vazduha pored leptira imaju izražen pulsirajući karakter i pri najmirnijem radu motora na ustaljenom režimu, kako je to pokazano u Odeljku 5.4.3.

Efekat pulsacija uveden je uz pomoć Matlabove funkcije *rand* za generisanje pseudoslučajnih brojeva iz intervala  $[0,1]$  sa uniformnom raspodelom. Da bi se efekat pulsacija ugradio u simulacije neispravnosti, a time i u dijagnostički model, napisane su dve vrlo slične funkcije koje će to omogućiti: *generisiPulsacije.m* i *generisiPulsacijeProc.m*. Razlika između ove dve funkcije je samo u tome, što se u prvoj zona pulsacija definiše u apsolutnim veličinama, a u drugoj procentualno. Obema funkcijama prosleđuju se dva parametra:

- jedna kolona matrice rezultata sa konstantnom ustaljenom vrednošću veličine kojoj se želi dodati efekat pulsiranja (broj obrtaja, pritisak u usisnom kolektoru ili maseni protok vazduha), označena sa *MRkol* i
- maksimalno željeno odstupanje tretirane veličine u odnosu na ustaljenu vrednost iz kolone matrice rezultata (označeno sa *amp* u funkciji *generisiPulsacije*, a sa *kProc* u funkciji *generisiPulsacijeProc*).

Kao rezultat bilo koje od ove dve funkcije dobija se obrađena kolona matrice rezultata sa pulsirajućim vrednostima odabrane veličine, u definisanim granicama. Grubi pseudokodovi koji predstavljaju suštinu ovih funkcija dati su u nastavku

**Funkcija generisiPulsacije(MRkol, amp)**

postavi donju granicu pulsacije:

$$\text{donjaGranica} = \text{MRkol} - \text{amp}$$

odredi širinu intervala pulsiranja:

$$\text{širinaIntervala} = 2 \cdot \text{amp}$$

generiši pseudoslučajne brojeve iz intervala pulsiranja:

$$\text{nizSlučajnihBrojeva} = \text{širinaIntervala} \cdot \text{rand}$$

generiši pulsirajuću kolonu:

$$\text{pulsirajućaKolona} = \text{donjaGranica} + \text{nizSlučajnihBrojeva}$$

**kraj**

**Funkcija generisiPulsacijeProc(MRkol, kProc)**

postavi donju granicu pulsacije:

$$\text{donjaGranica} = \text{MRkol} \cdot (100 - \text{kProc}) / 100$$

odredi širinu intervala pulsiranja:

$$\text{širinaIntervala} = \text{MRkol} \cdot 2 \cdot \text{kProc} / 100$$

generiši pseudoslučajne brojeve iz intervala pulsiranja:

$$\text{nizSlučajnihBrojeva} = \text{širinaIntervala} \cdot \text{rand}$$

generiši pulsirajuću kolonu:

$$\text{pulsirajućaKolona} = \text{donjaGranica} + \text{nizSlučajnihBrojeva}$$

**kraj**

Definisanje intervala, u kojem će konkretna izlazna veličina simulatora neispravnosti pulsirati, izvršeno je na osnovu praćenja promena izmerenih vrednosti tih veličina u vremenu kod stvarnog motora, kao na Sl. 5.7 i Sl. 5.8, ali na većem broju ustaljenih režima rada. Na osnovu toga, odabrani su sledeći parametri za definisanje intervala pulsiranja:

- za broj obrtaja motora,  $\text{amp} = 10 \text{ min}^{-1}$ ,
- za pritisak u usisnom kolektoru,  $\text{kProc} = 2\%$  i
- za maseni protok vazduha pored leptira,  $\text{kProc} = 0.8\%$ .

Drugim rečima, pomoću funkcija *generisiPulsacije* i *generisiPulsacijeProc* broj obrtaja motora varira se u granicama  $\pm 10 \text{ min}^{-1}$ , pritisak u usisnom kolektoru u granicama  $\pm 2\%$ , a maseni protok vazduha pored leptira u granicama  $\pm 0.8\%$ , u odnosu na svoje konstantne vrednosti dobijene pomoću modela **NSimulator**.

Obradom matrice *MatricaRezultata* na način opisan u ovom odeljku, dobija se izlazna matrica simulatora neispravnosti,  $\text{MatBRT}(\alpha_{\text{PROC}}, T_a, SP)$ , čija je struktura ilustrovana ranije na Sl. 5.5.

### **Korak VII: Iscrtavanje dijagrama**

Izlazna matrica *MatBRT* sadrži suštinske podatke koji su neophodni za razvoj dijagnostičkog modela za otkrivanje i prepoznavanje neispravnosti. Međutim, pre nego što ti podaci budu iskorišćeni u dijagnostičke svrhe, neophodno je vizuelno ih sagledati zbog utvrđivanja eventualnih nelogičnosti u rezultatima simulacije, ali i zbog definisanja načina organizovanja podataka tokom razvoja dijagnostičkog modela. To se najlakše može uraditi,



ako se ključne kolone matrice *MatBRT* predstave dijagramima. Odavde proizilazi da na osnovu matrice *MatBRT* treba nacrtati po 3 ili 4 dijagrama, što ne bi trebalo da bude neki problem. Problem se pojavljuje, kada se uzme u obzir da su u jednoj matrici *MatBRT* samo podaci koji se odnose na jednu baznu radnu tačku motora, a dijagnostički model zahteva rezultate simulacija u relativno velikom broju baznih radnih tačaka. Osim toga, u velikom broju baznih radnih tačaka motora treba izvršiti i simulacije kojima se generišu podaci za testiranje i proveru robusnosti dijagnostičkog modela. Time se broj dijagrama, koje treba iscrtati, znatno uvećava. A sve ovo treba uraditi pri radu motora u zatvorenoj i u otvorenoj sprezi sa lambda-sondom, pa još i za dve vrste motora – sa masenim protokomerom i bez protokomera. Kad se sve ovo uzme u obzir, potrebno je nacrtati ogroman broj dijagrama kako bi se sagledali rezultati simulacija neispravnosti. Kasnije će biti date preciznije brojke, a ovde se, samo radi ilustracije problema, daje približan broj – trebalo je iscrtati i označiti preko 2000 dijagrama koji ilustruju rezultate svih simulacija neispravnosti.

Opisani problem je uslovio da se automatizuje i ovaj značajan segment u razvoju dijagnostičkog modela. Otuda je napisan poseban program pod nazivom *iscrtajDijagrame*, koji omogućava celoviti, a prilično brz uvid u rezultate svih simulacija neispravnosti. Ovaj program se poziva u okviru *NSimulatorMenadzer*-a, a listing njegovog izvornog koda dat je u prilogu P2.4.

### **Korak VIII: Snimanje izlaznih rezultata**

Izlazne rezultate programa *NSimulatorMenadzer* predstavljaju matrica rezultata simulacija neispravnosti u jednoj baznoj radnoj tački, *MatBRT*, kao i prateći dijagrami sa vizuelnom interpretacijom simuliranih neispravnosti. Zbog potrebe za kasnijim korišćenjem ovih izlaznih rezultata, oni su morali da se čuvaju u odgovarajućim fajlovima. Iz razloga objašnjenih u prethodnom koraku, broj ovih fajlova je izuzetno veliki, pa se morala osmisliti procedura za automatsko generisanje deskriptivnih imena za svaki od njih. Ova procedura je implementirana u okviru programa *NSimulatorMenadzer*. Na taj način, omogućeno je relativno jednostavno prepoznavanje sadržaja fajla na osnovu njegovog naziva.

Kao ilustracija, ovde će biti naveden naziv jednog Excel fajla, u koji je snimljena jedna konkretna matrica *MatBRT*. Na primer, automatski generisani naziv fajla

„MAF-PS - Sim25-0.25s - a08 - Brz.IV - Temp.30,Lep.12%.xls“,  
sadrži sledeće informacije:

- da su u njemu rezultati simulacija neispravnosti na motoru sa masenim protokomerom vazduha (MAF),
- da se simulacije odnose na rad motora u povratnoj sprezi sa lambda-sondom (PS),
- da je vreme jedne simulacije 25 sekundi,
- da je interval uzorkovanja 0.25 sekundi,
- da ugao leptira u zatvorenom položaju iznosi 8 stepeni (a08),
- da se simulacije odnose na baznu radnu tačku definisanu otvorom leptira 12%, četvrtim stepenom prenosa menjačkog prenosa i temperaturom okolnog vazduha 30°C.

S obzirom da broj fajlova koji su bili neophodni u istraživanju iznosi preko 2500, jasno je da je automatizovanim dodeljivanjem imena obezbeđeno relativno lako pronalaženje odgovarajućih podataka u kasnijem postupku generisanja dijagnostičkog modela.

U *Tab. 5.7*, samo radi ilustracije, prikazan je deo podataka koje sadrži baš gore pomenuti Excel fajl. Prva kolona i zaglavlje iz *Tab. 5.7* inače ne postoje u izvornom fajlu, ali su ovde pridodati zbog preglednosti. Treba još napomenuti da su, osim podataka o vrednostima veličina  $p_i$ ,  $T_i$  i  $\dot{m}_{at}$  koje su očitali odgovarajući senzori, u jednom ovakvom excel fajlu sadržani još i podaci o stvarnim vrednostima ovih veličina. Njihove stvarne vrednosti

prikazuju se u dijagramima, u svrhu poređenja, ali se ne koriste pri generisanju dijagnostičkog modela, pa zato ovde nisu prikazane.

Tab. 5.7 Ilustracija podataka sadržanih u izlaznoj matrici *MatBRT*

No	$t$ [s]	$\alpha_{\text{PROC}}$ [%]	$n$ [ $\text{min}^{-1}$ ]	$p_i$ [bar]	$T_i$ [K]	$T_a$ [K]	$\dot{m}_{\text{at}}$ [kg/s]
1	22	12	2727.086	0.587217	332.0	303	0.012489
2	22.25	12	2725.771	0.580123	332.0	303	0.012540
3	22.5	12	2723.295	0.587802	332.0	303	0.012462
4	22.75	12	2710.593	0.587464	332.0	303	0.012518
5	23	12	2715.847	0.578151	332.0	303	0.012522
6	23.25	12	2718.780	0.583031	332.0	303	0.012499
7	23.5	12	2721.302	0.592017	332.0	303	0.012544
8	23.75	12	2715.262	0.590089	332.0	303	0.012538
9	24	12	2722.873	0.585189	332.0	303	0.012521
10	24.25	12	2723.104	0.588995	332.0	303	0.012506
11	24.5	12	2713.779	0.583543	332.0	303	0.012541
12	24.75	12	2722.228	0.583824	332.0	303	0.012477
13	25	12	2718.208	0.585037	332.0	303	0.012513
14	22	12	2717.891	0.477874	332.0	303	0.012477
15	22.25	12	2722.570	0.483142	332.0	303	0.012544
16	22.5	12	2719.972	0.489865	332.0	303	0.012497
17	22.75	12	2718.655	0.490955	332.0	303	0.012511
18	23	12	2725.096	0.484137	332.0	303	0.012524
.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.
173	22.75	12	2741.901	0.541722	320.4	303	0.012766
174	23	12	2745.550	0.535901	320.4	303	0.012837
175	23.25	12	2740.582	0.541046	320.4	303	0.012779
176	23.5	12	2747.396	0.541864	320.4	303	0.012791
177	23.75	12	2741.339	0.537002	320.4	303	0.012759
178	24	12	2744.628	0.538127	320.4	303	0.012813
179	24.25	12	2749.603	0.540525	320.4	303	0.012801
180	24.5	12	2749.872	0.541997	320.4	303	0.012796
181	24.75	12	2745.148	0.534885	320.4	303	0.012780
182	25	12	2745.079	0.547848	320.4	303	0.012773

## 5.5 Rezultati primene simulatora neispravnosti

Zbog izuzetno velikog broja izvršenih simulacija u cilju pripreme podataka za razvoj dijagnostičkog modela, nemoguće je prikazati rezultate svih simulacija. Iz tog razloga, u ovom odeljku biće prikazani rezultati simulacija neispravnosti u nekoliko karakterističnih slučajeva, za motor bez protokomera i za motor sa masenim protokomerom, kako bi se prikazale mogućnosti razvijenog simulatora neispravnosti. Svi karakteristični slučajevi odnose se na istu baznu radnu tačku, kako bi se lakše uočile specifičnosti svakog od njih. Konkretna bazna radna tačka definisana je sledećim vrednostima relevantnih parametara:

- procentualni otvor leptira  $\alpha_{\text{PROC}} = 12\%$ ,
- stepen prenosa menjačkog prenosnika  $SP = IV$  i
- temperatura okoline  $T_a = 283\text{K}$ .

Posmatrana bazna radna tačka je odabrana iz skupa baznih radnih tačaka predviđenih za razvoj dijagnostičkog modela. O načinu formiranja ovog skupa biće reči u Poglavlju 6.

U nastavku će biti prikazani dijagrami koji pokazuju kako se menjaju izlazne veličine simulatora neispravnosti ( $p_i$ ,  $n$ ,  $T_i$  i  $\dot{m}_{at}$ ) pri radu motora sa svakom od 13 razmatranih neispravnosti, u odnosu na ispravno stanje. Pored ovih, osnovnih dijagrama, biće prikazani još neki pomoćni dijagrami, koji će omogućiti lakše tumačenje rezultata simulacija. To su dijagrami masenih protoka ( $\dot{m}_{fv}$ ,  $\dot{m}_{ap}$ ,  $\dot{m}_{EGR}$ ) i dijagrami željene, nekorigovane i konačne vrednosti koeficijenta viška vazduha ( $\lambda_{zelj}$ ,  $\lambda_{nkr}$ ,  $\lambda_{kon}$ ). Veličine  $\dot{m}_{fv}$ ,  $\dot{m}_{ap}$  i  $\dot{m}_{EGR}$  su određene iz jednačina (5.28), (5.32) i (5.39), respektivno,  $\lambda_{nkr}$  prema jednačini (5.62),  $\lambda_{kon}$  prema algoritmu sa Sl. 5.4, a sve je implementirano u Simulink modelu **NSimulator**.

Kod svih dijagrama, na apscisi je broj simulacionih uzoraka, koji je jednak broju vrsta matrice MatBRT, tj. iznosi 182. Dijagrami su podeljeni na 14 vertikalnih zona i svaka zona odgovara jednom stanju usisnog sistema. U svakoj zoni dijagrama nalazi se oznaka stanja na koje se ta zona odnosi, a tumačenje svake oznake dato je u Tab. 5.8.

Tab. 5.8 Objašnjenje oznaka stanja usisnog sistema

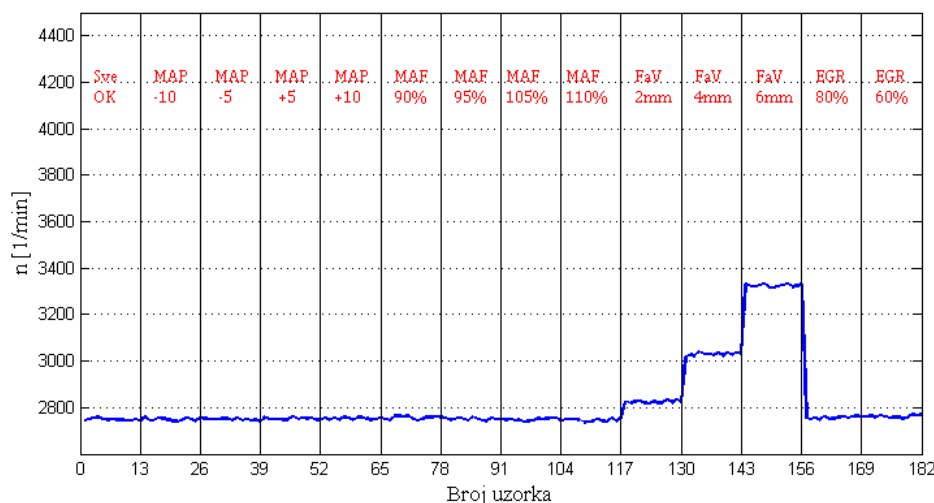
Oznaka stanja usisnog sistema	Naziv stanja usisnog sistema
Sve OK	Nema neispravnosti
MAP -10	Senzor pritiska u usisnom kolektoru (MAP senzor) očitava za 10 kPa nižu vrednost
MAP -5	Senzor pritiska u usisnom kolektoru (MAP senzor) očitava za 5 kPa nižu vrednost
MAP +5	Senzor pritiska u usisnom kolektoru (MAP senzor) očitava za 5 kPa višu vrednost
MAP +10	Senzor pritiska u usisnom kolektoru (MAP senzor) očitava za 10 kPa višu vrednost
IAT -15	Senzor temperature u usisnom kolektoru (IAT senzor) očitava za 15° nižu vrednost
IAT -10	Senzor temperature u usisnom kolektoru (IAT senzor) očitava za 10° nižu vrednost
IAT +10	Senzor temperature u usisnom kolektoru (IAT senzor) očitava za 10° višu vrednost
IAT +15	Senzor temperature u usisnom kolektoru (IAT senzor) očitava za 15° višu vrednost
MAF 90%	Senzor masenog protoka vazduha (MAF senzor) očitava 10 % nižu vrednost
MAF 95%	Senzor masenog protoka vazduha (MAF senzor) očitava 5 % nižu vrednost
MAF 105%	Senzor masenog protoka vazduha (MAF senzor) očitava 5 % višu vrednost
MAF 110%	Senzor masenog protoka vazduha (MAF senzor) očitava 10 % višu vrednost
FaV 2mm	Ulaz „falš“-vazduha kroz otvor prečnika 2mm
FaV 4mm	Ulaz „falš“-vazduha kroz otvor prečnika 4mm
FaV 6mm	Ulaz „falš“-vazduha kroz otvor prečnika 6mm
EGR 80%	EGR protok umanjen na 80%
EGR 60%	EGR protok umanjen na 60%

### 5.5.1 Motor sa masenim protokomerom vazduha

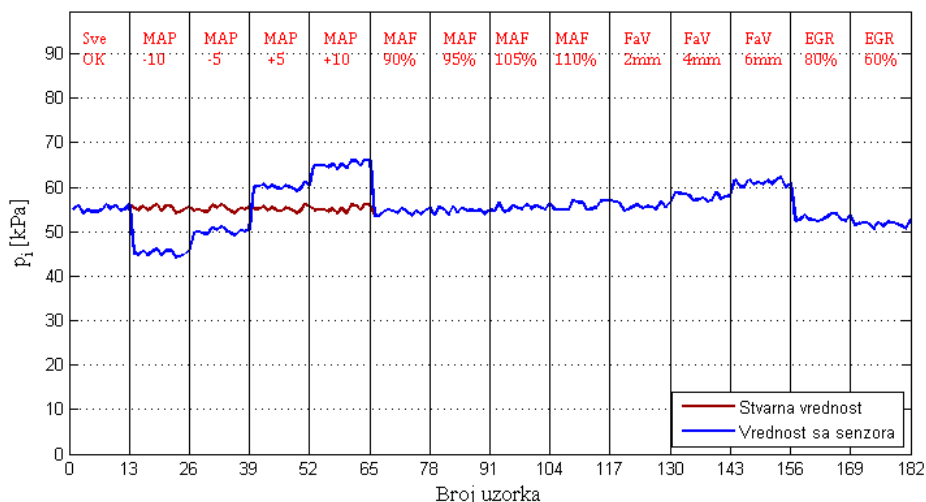
#### Slučaj 1.1: Rad u zatvorenoj sprezi sa $\lambda$ -sondom, korekcija vrednosti $\lambda$ nije ograničena

Rezultati simulacija neispravnosti u usisnom sistemu motora sa masenim protokomerom, za slučaj kada ne postoji ograničenje korekcije koeficijenta viška vazduha, prikazani su na Sl. 5.11 do Sl. 5.16.

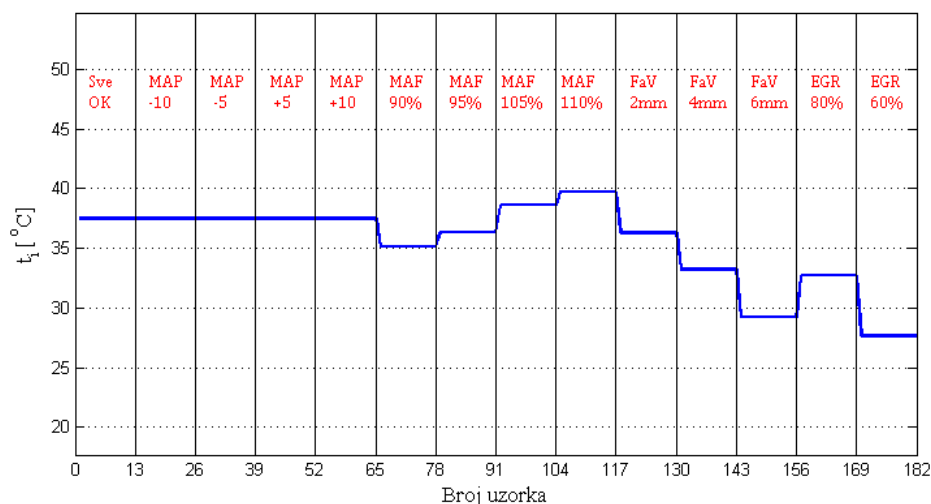
Na osnovu ovih rezultata, može se zapaziti kakav uticaj pojedine neispravnosti imaju na posmatrane karakteristične veličine motora. U nastavku se taksativno daju detalji ovih zapažanja sa odgovarajućim objašnjenjima.



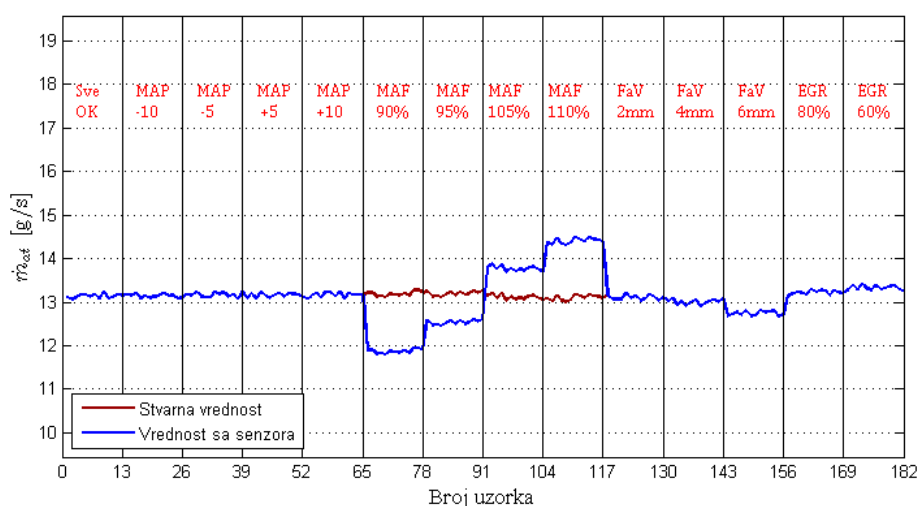
Sl. 5.11 Promena broja obrtaja motora pri simulacijama neispravnosti kod motora sa masenim protokomerom vazduha – Slučaj 1.1



Sl. 5.12 Promena pritiska u usisnom kolektoru pri simulacijama neispravnosti kod motora sa masenim protokomerom vazduha – Slučaj 1.1



Sl. 5.13 Promena temperature u usisnom kolektoru pri simulacijama neispravnosti kod motora sa masenim protokomerom vazduha – Slučaj 1.1



Sl. 5.14 Promena masenog protoka pored leptira pri simulacijama neispravnosti kod motora sa masenim protokomerom vazduha – Slučaj 1.1

### Neispravnost: Pogrešno očitavanje senzora pritiska u usisnom kolektoru (MAP -10, MAP -5, MAP +5, MAP +10)

**Zapažanja:** Ova neispravnost nema primetnijeg uticaja na posmatrane promenljive kod motora sa masenim protokomerom. Vrednosti svih posmatranih promenljivih ostaju na nivou njihovih vrednosti kod ispravnog motora. Jedino što se razlikuje u odnosu na ispravan motor je vrednost pritiska sa senzora, kako se vidi na Sl. 5.12. Ova vrednost je manja ili veća za 5 ili 10 kPa od stvarne vrednosti, u zavisnosti od konkretne neispravnosti.

**Objašnjenja:** Kod ove vrste motora, senzor pritiska u usisnom kolektoru nema presudnu ulogu u doziranju goriva. Uglavnom služi za otkrivanje neispravnosti na drugim senzorima i sistemima (najčešće na EGR ventilima) ili kao rezervna opcija za određivanje opterećenja motora, kada otkáže maseni protokomer vazduha [111]. To su razlozi zbog kojih ova neispravnost ne dovodi do primetnijih poremećaja posmatranih veličina.

## **Neispravnost: Pogrešno očitavanje senzora masenog protoka vazduha (MAF 90%, MAF 95%, MAF 105%, MAF 110%)**

Zapažanja: Ova neispravnost dovodi do promena nekih posmatranih promenljivih kod motora sa masenim protokomerom, dok su druge bez osetnijih promena.

Kako se vidi na Sl. 5.11, vrednost broja obrtaja ostaje na nivou vrednosti kod ispravnog motora.

Pritisak u usisnom kolektoru pokazuje trend vrlo blagog pada u odnosu na ispravan motor, kada su u pitanju greške očitavanja niže vrednosti (MAF 90% i MAF 95%), a obrnuti trend, kada se očitavaju više vrednosti masenog protoka vazduha (MAF 105% i MAF 110%). Ova konstatacija je ilustrovana na Sl. 5.12.

Kod grešaka MAF 90% i MAF 95%, temperatura u usisnom kolektoru je niža u odnosu na istu kod ispravnog motora. Obrnuto, pri greškama MAF 105% i MAF 110%, temperatura u usisnom kolektoru raste u odnosu na stanje bez neispravnosti. Što je veća neispravnost, posledice po temperaturu u usisnom kolektoru su sve izraženije, što se vidi na Sl. 5.13.

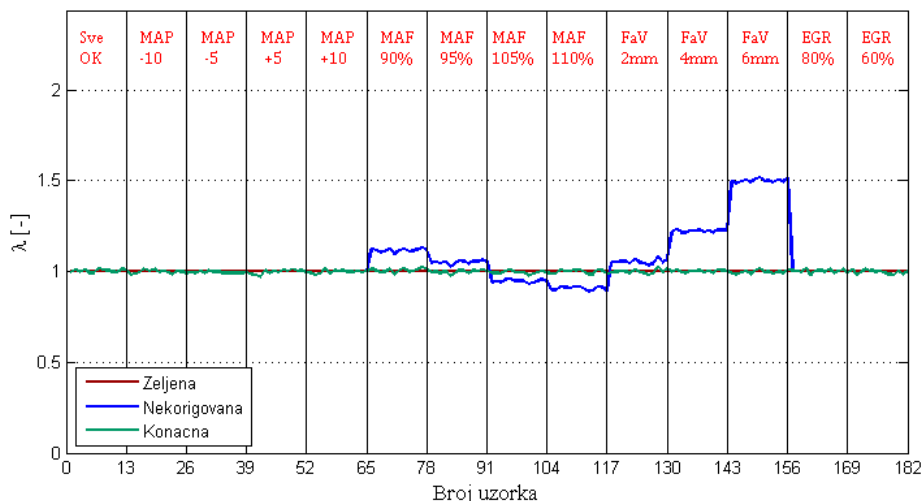
Očitavanje senzora masenog protoka vazduha je manje ili veće za 5 ili 10 procenata od stvarne vrednosti, u zavisnosti od konkretne neispravnosti (Sl. 5.14). Primećuju se i vrlo, vrlo blage promene stvarnog masenog protoka pored leptira u odnosu na ispravan motor i to naviše, pri greškama MAF 90% i MAF 95%, a naniže pri greškama MAF 105% i MAF 110%.

Objašnjenja: Kod ove vrste motora, očekuje se da pogrešno očitavanje MAF senzora ima uticaja na rad motora, jer je njegov signal veoma bitan za doziranje goriva. Na Sl. 5.15 vidi se kakav je uticaj ove neispravnosti na vrednost koeficijenta viška vazduha. Primećuje se da niža očitana vrednost masenog protoka (MAF 90% i MAF 95%) dovodi do povećanja koeficijenta viška vazduha u odnosu na željenu vrednost (plava linija dijagrama na Sl. 5.15). Ovo je logično, jer ECU smanjuje dozu goriva „misleći“ da manje vazduha ulazi u motor, a zapravo se protok vazduha nije promenio ili se neznatno promenio. Obrnuta je situacija kod neispravnosti MAF 105% i MAF 110%, što bi trebalo da dovede do obogaćenja smeše. Međutim, pošto u ovom slučaju motor radi u režimu povratne sprege sa  $\lambda$ -sondom, dolazi do odgovarajuće korekcije doze goriva, kojom se konačna vrednost koeficijenta viška vazduha dovodi na željeni nivo. Drugim rečima, povratna sprega sa  $\lambda$ -sondom praktično ispravlja grešku očitavanja masenog protokomera, zadržavajući broj obrtaja na nivou njegovih vrednosti kod ispravnog motora.

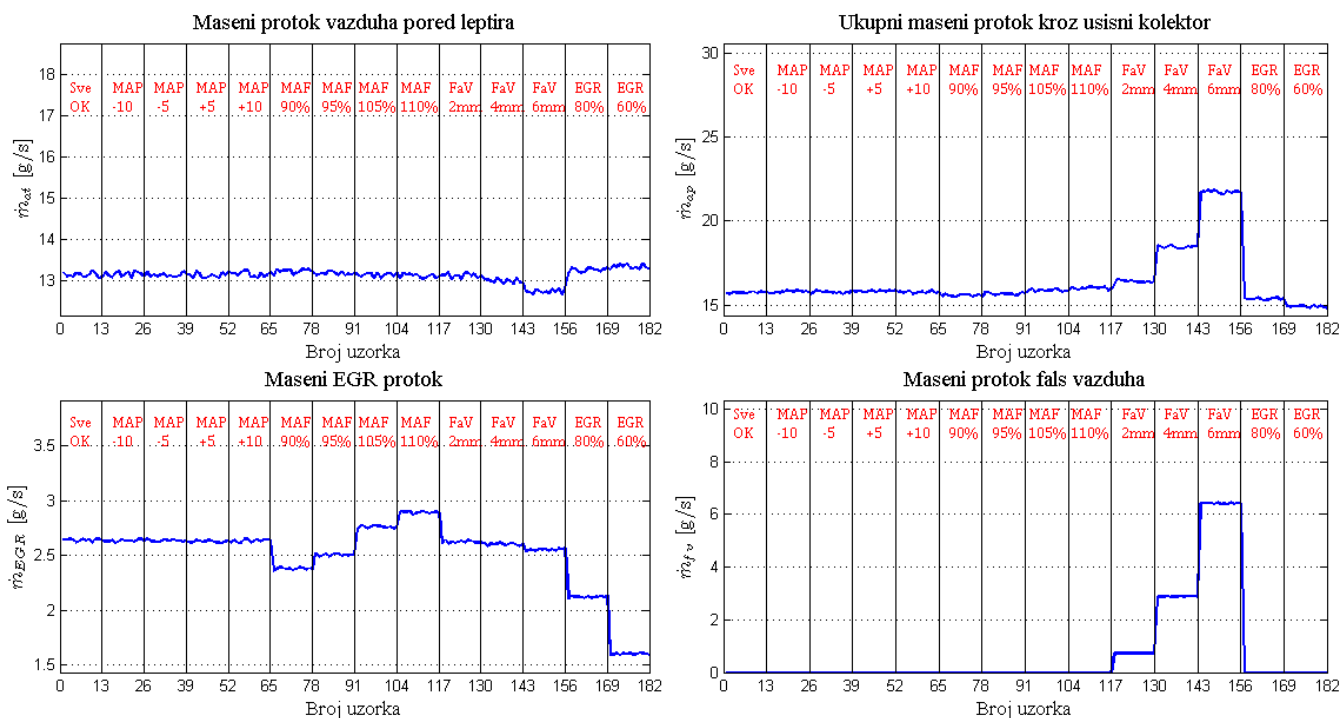
Promene temperature u usisnom kolektoru nastaju zbog promena masenog EGR protoka, jer je temperatura recirkulisanih izduvni gasova veća od temperature u usisnom kolektoru. Na Sl. 5.16, na dijagramu masenog EGR protoka, vidi se da on raste od neispravnosti MAF 90% ka neispravnosti MAF 110%. Ovo je posledica pretpostavke da ECU dozira EGR protok proporcionalno signalu sa MAF senzora. To znači da niža očitana vrednost masenog protoka vazduha u odnosu na stvarnu, dovodi do smanjenja EGR protoka i obrnuto, viša vrednost povećava EGR protok. Dalje, zbog nepromenjenog broja obrtaja, promena EGR protoka ima određenog uticaja na stvarni protok pored leptira i to tako što smanjenje EGR protoka uzrokuje veoma blago uvećanje protoka pored leptira i obrnuto, porast EGR protoka uzrokuje veoma blago umanjeње protoka pored leptira. Imajući sve ovo u vidu, postaje jasno zbog čega je temperatura u usisnom kolektoru pri neispravnostima MAF 90% i MAF 95% niža, a pri neispravnostima MAF 105% i MAF 110% viša od odgovarajuće temperature u stanju bez neispravnosti.

Promene pritiska u usisnom kolektoru objašnjava pogled na dijagram ukupnog masenog protoka kroz usisni kolektor, na Sl. 5.16. Naime, posmatrajući od

neispravnosti MAF 90% ka neispravnosti MAF 110%, ukupni maseni protok kroz usisni kolektor ima veoma blagi rast. Pritisak u usisnom kolektoru mora imati isti takav trend, s obzirom na činjenicu da broj obrtaja motora ostaje nepromenjen.



Sl. 5.15 Karakteristične vrednosti koeficijenta viška vazduha pri simuliranju neispravnosti kod motora sa masenim protokomerom vazduha – Slučaj 1.1



Sl. 5.16 Relevantni maseni protoci pri simuliranju neispravnosti kod motora sa masenim protokomerom vazduha – Slučaj 1.1

**Neispravnost: Ulaz „fals“-vazduha u usisni sistem motora (FaV 2mm, FaV 4mm, FaV 6mm)**

**Zapažanja:** Ova neispravnost utiče na posmatrane promenljive kod ove vrste motora. Broj obrtaja motora i pritisak u usisnom kolektoru se povećavaju, dok se maseni protok pored leptira i temperatura u usisnom kolektoru smanjuju. Ove promene su utoliko izraženije, ukoliko je veći otvor kroz koji struji „fals“-vazduh.

**Objašnjenja:** Povećanje broja obrtaja na prvi pogled nije logično kod ove vrste motora, jer „falš“-vazduh ulazi u motor bez „znanja“ elektronske upravljačke jedinice. To bi trebalo da dovede do osiromašenja smeše, kako pokazuje *Sl. 5.15*, a time i do pada snage i broja obrtaja motora. Međutim, u ovom slučaju motor radi u povratnoj sprezi sa  $\lambda$ -sondom, pa će signal  $\lambda$ -sonde o siromašnoj smeši uticati na povećanje doze goriva do nivoa, koji odgovara željenoj vrednosti koeficijenta viška vazduha. Otuda će povećana količina sveže smeše željenog sastava dovesti do povećanja snage, a time i broja obrtaja motora.

Smanjenje masenog protoka vazduha pored leptira i povećanje pritiska u usisnom kolektoru mogu se objasniti pomoću *Sl. 5.16*. Na ovoj slici vidi se da dolazi do povećanja ukupnog protoka kroz usisni kolektor, što objašnjava porast pritiska u njemu, bez obzira na porast broja obrtaja. Blagi pad masenog protoka vazduha pored leptira posledica je punjenja usisnog kolektora „falš“-vazduhom.

Temperatura u usisnom kolektoru niža je iz dva razloga. Osnovni razlog je taj, što je temperatura „falš“-vazduha niža od temperature u usisnom kolektoru, pa dotok tog neželjenog vazduha ima rashlađujući efekat. Drugi, znatno slabiji uticaj je blago smanjenje EGR protoka, do kojeg dolazi usled dotoka „falš“-vazduha. Ovo se može uočiti na dijagramu masenog EGR protoka, na *Sl. 5.16*.

### **Neispravnost: Umanjen EGR protok (EGR 80%, EGR 60%)**

**Zapažanja:** Ova neispravnost ima uticaja na neke posmatrane promenljive kod ove vrste motora. Kako se vidi na *Sl. 5.12* i *Sl. 5.13*, pritisak i temperatura u usisnom kolektoru su niži u odnosu na njihove vrednosti kod ispravnog motora. Što je veće umanjenje EGR protoka, ove posledice su izraženije. Maseni protok pored leptira je neznatno viši nego kod ispravnog motora (*Sl. 5.14*), a na sličan način, u još blažoj formi, menja se i broj obrtaja (*Sl. 5.11*).

**Objašnjenja:** Do pada temperature dolazi zbog manjeg dotoka recirkulisanih izduvni gasova, čija temperatura je znatno viša od temperature u usisnom kolektoru. Otuda će zagrevanje sredine u usisnom kolektoru biti slabije.

Blago povećanje masenog protoka pored leptira uzrokovano je smanjenjem dotoka EGR, što je obrnuti efekat u odnosu na slučaj ulaska „falš“-vazduha u usisni kolektor. Razlog nižeg pritiska u usisnom kolektoru može se pronaći na *Sl. 5.16*. Tamo se vidi da je ukupni maseni protok kroz usisni kolektor manji, nego kod ispravnog motora, što znači da je pad EGR protoka izraženiji od porasta protoka pored leptira.

Vrlo blago povećanje masenog protoka pored leptira dovodi do odgovarajućeg povećanja doze goriva, što sve zajedno daje veoma, veoma blago povećanje snage motora. Otuda i neznatno veći broj obrtaja u odnosu na isti kod ispravnog motora.

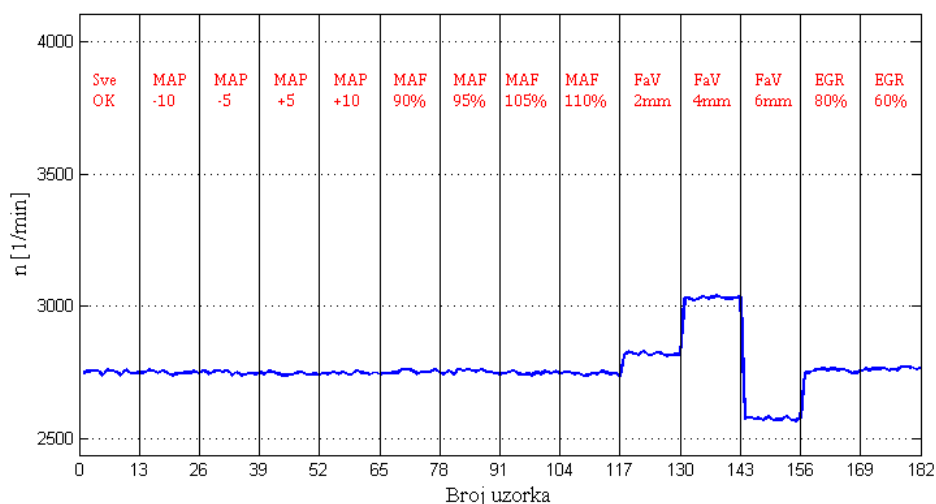
### **Slučaj 1.2: Rad u zatvorenoj sprezi sa $\lambda$ -sondom, korekcija vrednosti $\lambda$ ograničena**

Ovaj slučaj se ne razlikuje mnogo od Slučaja 1.1, jer je takođe u pitanju rad motora u zatvorenoj sprezi sa lambda-sondom. Razlika se pojavljuje jedino kod neispravnosti koje dovode do većeg poremećaja koeficijenta viška vazduha. U tom slučaju se, korekcijom doziranja goriva, koeficijent viška vazduha ne može dovesti na nivo željene vrednosti, kako je objašnjeno u Odeljku 5.3.7.3. Maksimalna dozvoljena vrednost korekcije  $\Delta\lambda_{\text{doz}}$  ide do 25% [114], a u ovom istraživanju je usvojena vrednost 20%.

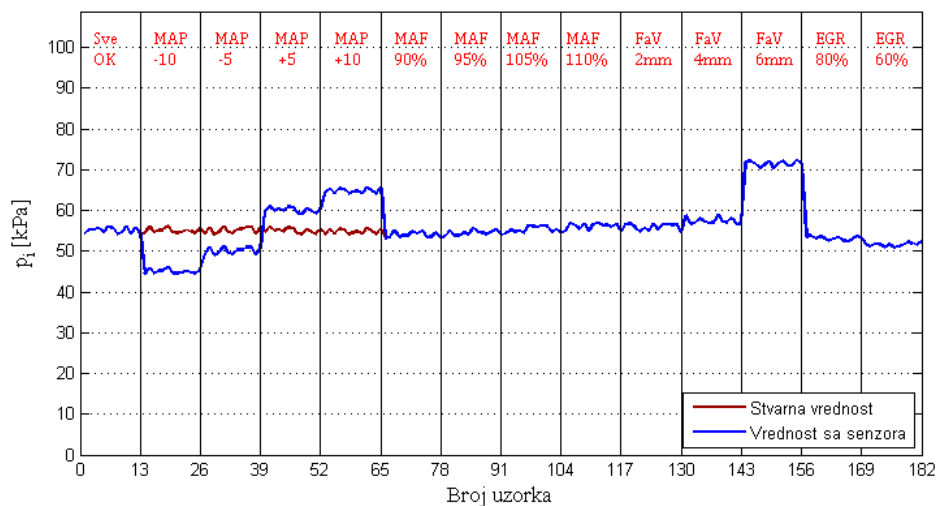
U nastavku će biti prikazani rezultati simulacija za ovaj slučaj, a onda će se u diskusiji pažnja posvetiti samo onim neispravnostima, koje ga čine drugačijim od Slučaja 1.1. Kod svih ostalih neispravnosti, zapažanja i objašnjenja bi bila potpuno ista, kao u Slučaju 1.1, pa su ovde izostavljena, da bi se izbeglo ponavljanje.



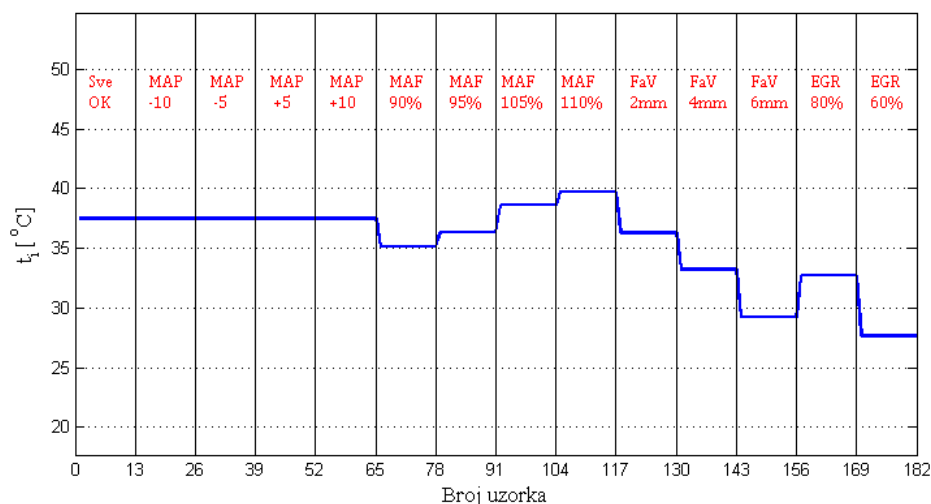
Na Sl. 5.17 - Sl. 5.20 prikazani su uticaji pojedinih simuliranih neispravnosti na posmatrane promenljive motora. Poređenjem Sl. 5.17 - Sl. 5.20 sa Sl. 5.11 - Sl. 5.14, respektivno, može se primetiti da se u odabranoj baznoj radnoj tački motora i za posmatrane intenzitete neispravnosti, Slučaj 1.2 razlikuje od Slučaja 1.1, samo po neispravnosti „Ulaz falš-vazduha kroz otvor prečnika 6mm (FaV 6mm)“. Zbog toga će ovde biti prodiskutovana samo ta neispravnost. Pri radu motora u nekim drugim baznim radnim tačkama i pri simuliranju nekih drugih intenziteta neispravnosti, verovatno bi se pojavile još neke razlike između Slučaja 1.2 i Slučaja 1.1, ali ovde sada nije cilj da se pronalaze te razlike, nego da se objasni situacija u kojoj ne može da se dostigne željena vrednost koeficijenta viška vazduha.



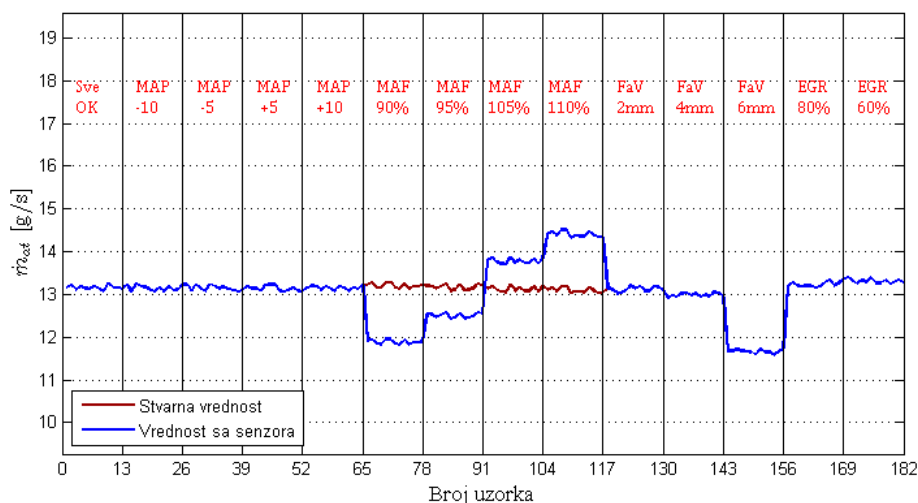
Sl. 5.17 Promena broja obrtaja motora pri simulacijama neispravnosti kod motora sa masenim protokomerom vazduha – Slučaj 1.2



Sl. 5.18 Promena pritiska u usisnom kolektoru pri simulacijama neispravnosti kod motora sa masenim protokomerom vazduha – Slučaj 1.2



Sl. 5.19 Promena temperature u usisnom kolektoru pri simulacijama neispravnosti kod motora sa masenim protokomerom vazduha – Slučaj 1.2



Sl. 5.20 Promena masenog protoka pored leptira pri simulacijama neispravnosti kod motora sa masenim protokomerom vazduha – Slučaj 1.2

### Neispravnost: Ulaz „fals“ -vazduha u usisni sistem kroz otvor prečnika 6 mm (FaV 6mm)

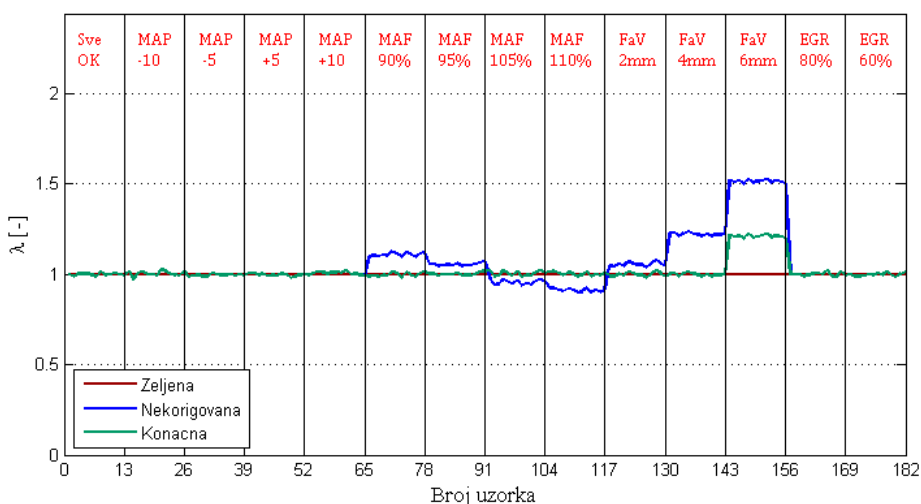
**Zapažanja:** Ova neispravnost značajno utiče na posmatrane promenljive motora. Broj obrtaja motora i maseni protok pored leptira su niži nego kod ispravnog motora, što je dijametralno drugačije ponašanje u odnosu na Slučaj 1.1. Pritisak u usisnom kolektoru je osetno viši, nego pri radu motora u ispravnom stanju, dok je u Slučaju 1.1 ovaj porast pritiska bio dosta umereniji. Temperatura u usisnom kolektoru je niža u odnosu na njenu vrednost u ispravnom stanju motora, kao u Slučaju 1.1.

**Objašnjenja:** Sva gore navedena zapažanja najjednostavnije se mogu objasniti pomoću dijagrama na Sl. 5.21, gde su prikazane karakteristične vrednosti koeficijenta viška vazduha dobijene simulacijama. Posmatranjem zone dijagrama obeležene oznakom „FaV 6mm“, primećuje se da bi usled relativno velikog dotoka neizmerenog vazduha trebalo da dođe do osetnog osiromašenja smeše (plava linija na dijagramu). Međutim, zbog zatvorene sprege sa  $\lambda$ -sondom, dolazi do korekcije doze goriva, ali samo u iznosu do 20%. Zato se vrednost  $\lambda$  sa nekorigovanih oko 1.5 svodi na konačnih (korigovanih) oko 1.2. To je razlog zbog čega dolazi do pada broja obrtaja u

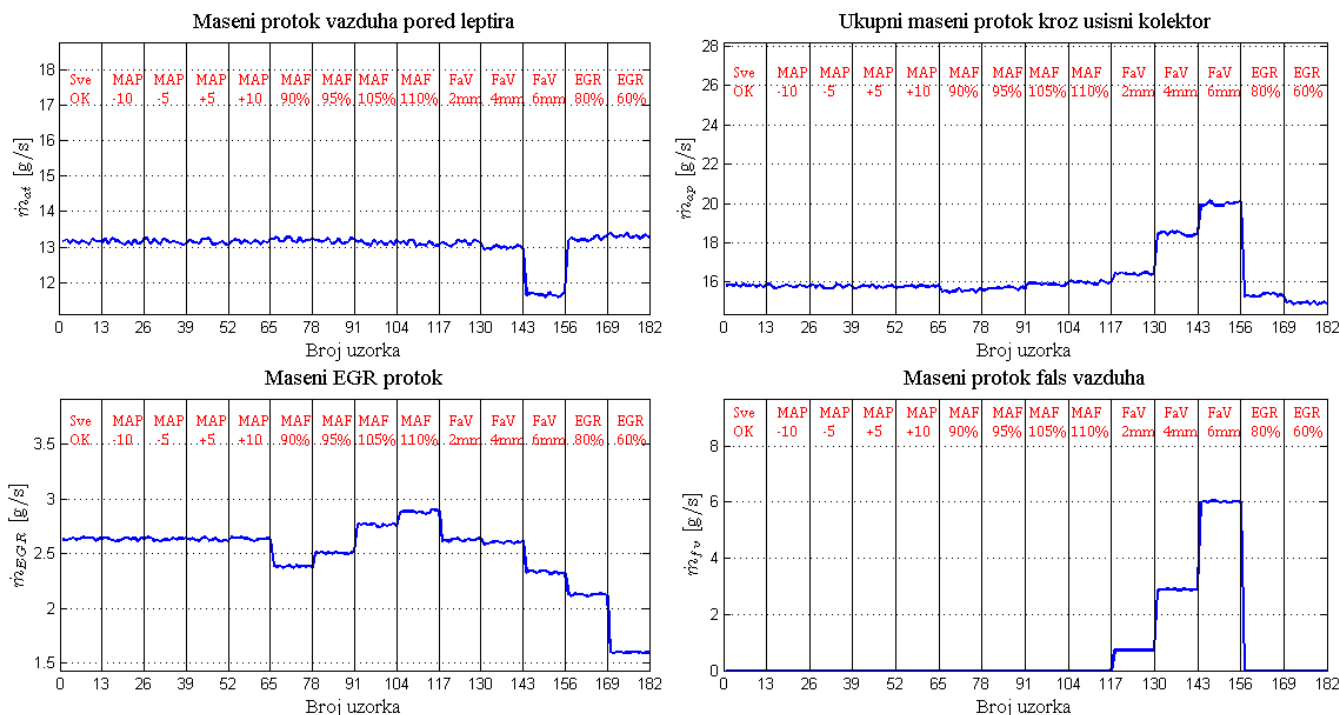
odnosu na stanje ispravnog motora, kada on radi sa vrednošću  $\lambda$  koja je bliska jedinici.

Pad masenog protoka pored leptira je posledica relativno velikog dotoka vazduha u usisni kolektor alternativnim putem. Sličan je uticaj „falš“-vazduha i na EGR protok, tako da je ukupan protok kroz usisni kolektor osetno manji nego u Slučaju 1.1, što se može uočiti poređenjem odgovarajućih dijagrama na Sl. 5.22 i Sl. 5.16.

Konačno, osetni porast pritiska u usisnom kolektoru posledica je kombinacije porasta ukupnog dotoka u usisni kolektor i umanjenja broja obrtaja motora, u odnosu na ispravno stanje.



Sl. 5.21 Karakteristične vrednosti koeficijenta viška vazduha pri simuliranju neispravnosti kod motora sa masenim protokomerom vazduha– Slučaj 1.2

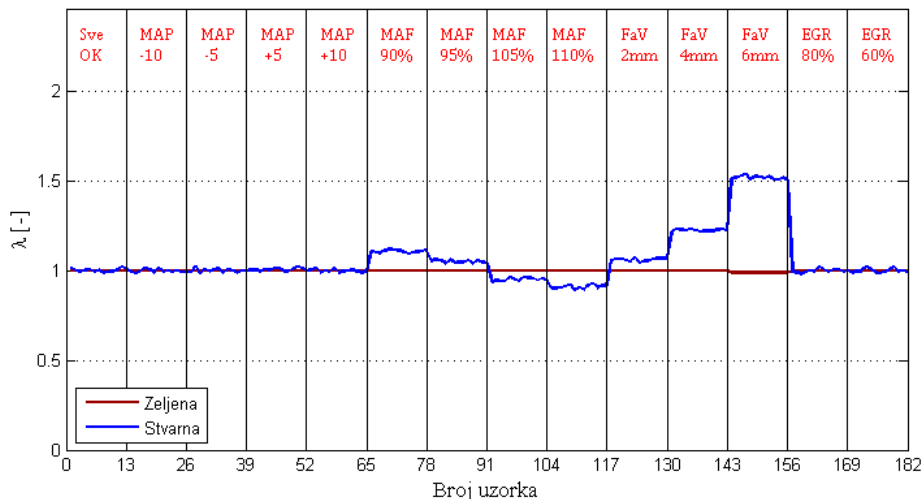


Sl. 5.22 Relevantni maseni protoci pri simuliranju neispravnosti kod motora sa masenim protokomerom vazduha– Slučaj 1.2

### Slučaj 1.3: Rad u otvorenoj sprezi sa $\lambda$ -sondom

Rezultati simulacija neispravnosti u usisnom sistemu motora sa masenim protokomerom, koji radi u režimu otvorene sprege sa lambda-sondom, prikazani su na Sl. 5.23 - Sl. 5.28.

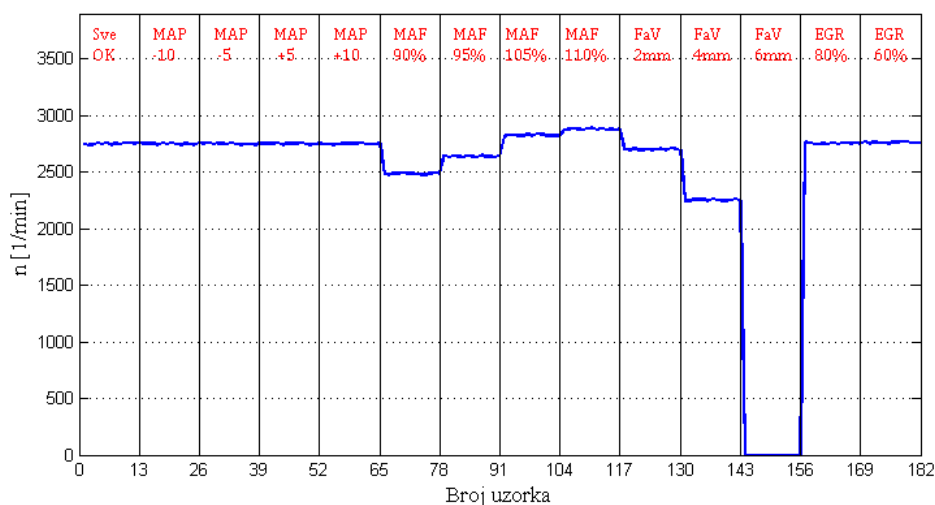
Ovde treba obratiti pažnju na neispravnosti koje dovode do odstupanja koeficijenta viška vazduha od željene vrednosti. Zbog nemogućnosti da se vrednost koeficijenta viška vazduha koriguje, ove neispravnosti će dovesti do promena vrednosti nekih posmatranih promenljivih u odnosu na ispravno stanje motora. Koje su to neispravnosti, najbolje se može videti sa dijagrama koeficijenta viška vazduha, prikazanog na Sl. 5.23.



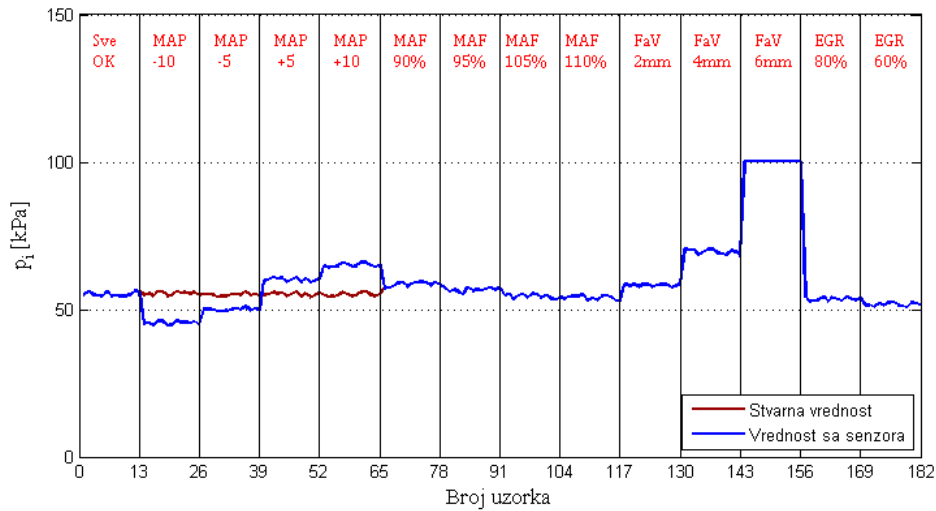
Sl. 5.23 Karakteristične vrednosti koeficijenta viška vazduha pri simuliranju neispravnosti kod motora sa masenim protokomerom vazduha – Slučaj 1.3

Kako pokazuje Sl. 5.23, neispravnosti „pogrešno očitavanje senzora pritiska u usisnom kolektoru“ i „umanjeni EGR protok“ ne dovode do poremećaja koeficijenta viška vazduha, pa bi zapažanja i objašnjenja bila identična onima u Slučaju 1.1. Da bi se izbeglo ponavljanje, komentari o ovim neispravnostima će ovde biti izostavljeni, a pažnja će biti posvećena neispravnostima „pogrešno očitavanje masenog protokomera vazduha“ i „ulaz falš-vazduha u usisni sistem“.

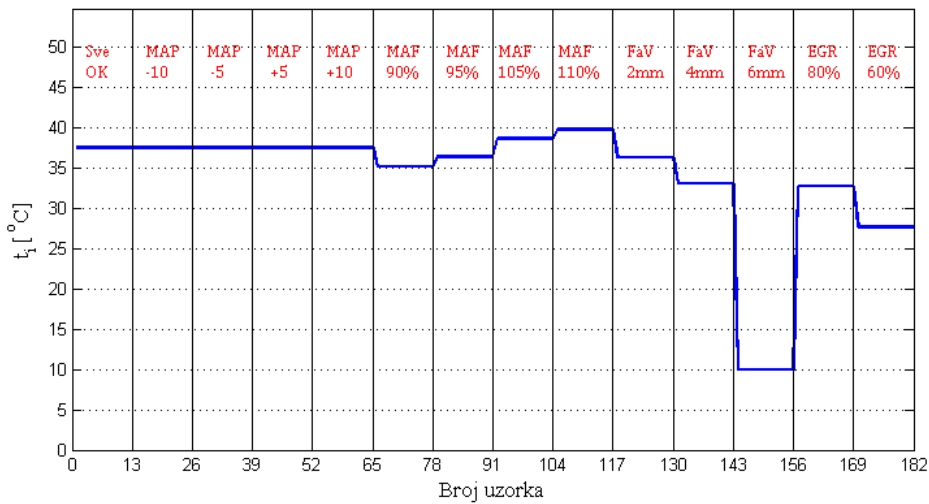
Na Sl. 5.24 - Sl. 5.27 prikazane su promene posmatranih karakterističnih promenljivih usled simuliranih neispravnosti, a kakve su to promene, biće prodiskutovano u nastavku.



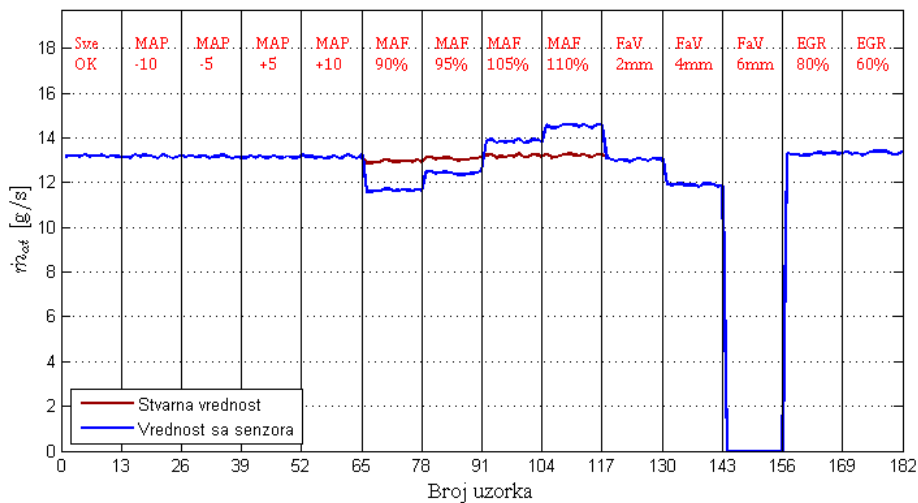
Sl. 5.24 Promena broja obrtaja motora pri simulacijama neispravnosti kod motora sa masenim protokomerom vazduha – Slučaj 1.3



Sl. 5.25 Promena pritiska u usisnom kolektoru pri simulacijama neispravnosti kod motora sa masenim protokomerom vazduha – Slučaj 1.3



Sl. 5.26 Promena temperature u usisnom kolektoru pri simulacijama neispravnosti kod motora sa masenim protokomerom vazduha – Slučaj 1.3



Sl. 5.27 Promena masenog protoka pored leptira pri simulacijama neispravnosti kod motora sa masenim protokomerom vazduha – Slučaj 1.3

**Neispravnost: Pogrešno očitavanje masenog protokomera vazduha (MAF 90%, MAF 95%, MAF 105%, MAF 110%)**

Zapažanja: Očitavanje manjeg masenog protoka (MAF 90% i MAF 95%) u odnosu na njegovu stvarnu vrednost, dovodi do umanjena broja obrtaja i vrlo blagog umanjena samog masenog protoka pored leptira, a pritisak u usisnom kolektoru postaje viši, u poređenju sa stanjem ispravnog motora. Ove posledice su sve izraženije, što je intenzitet neispravnosti veći.

Suprotno, očitavanje veće vrednosti masenog protoka (MAF 105% i MAF 110%) dovodi do povećanja broja obrtaja i vrlo blagog povećanja samog masenog protoka pored leptira, a pritisak u usisnom kolektoru postaje niži, u poređenju sa stanjem ispravnog motora. Ove posledice su izraženije kod veće greške očitavanja, ali se na osnovu *Sl. 5.24*, može primetiti da je broj obrtaja kod neispravnosti MAF 110% tek vrlo malo veći, nego kod neispravnosti MAF 105%. Otuda se može pretpostaviti da bi neka još veća greška očitavanja dovela do obrnutog trenda – pada broja obrtaja i masenog protoka pored leptira, a porasta pritiska u usisnom kolektoru.

Temperatura u usisnom kolektoru takođe se menja usled pogrešnog očitavanja masenog protokomera vazduha. Greške očitavanja niže vrednosti masenog protoka vazduha dovode do umanjena temperature, a greške očitavanja više vrednosti protoka uzrokuju porast temperature u usisnom kolektoru. Promene temperature su utoliko izraženije, ukoliko je greška očitavanja veća.

Objašnjenja: Greške MAF 90% i MAF 95% dovode do smanjenja doze goriva, tako da smeša postaje siromašnija (*Sl. 5.23*). Zato dolazi do umanjena snage motora, a time i do pada broja obrtaja. Usled manjeg broja obrtaja, vrednost pritiska u usisnom kolektoru je viša, nego kod ispravnog motora. Niži broj obrtaja je i razlog vrlo blagog pada masenog protoka vazduha.

Greške MAF 105% i MAF 110% uzrokuju povećanje doze goriva, a to znači i obogaćenje smeše, kako pokazuje *Sl. 5.23*. Zbog toga je snaga motora nešto veća, nego kod ispravnog motora, pa dolazi do blagog porasta broja obrtaja. Veći broj obrtaja dovodi do blagog pada pritiska u usisnom kolektoru, a takođe i do vrlo blagog porasta masenog protoka vazduha.

Činjenica da je porast broja obrtaja sve slabije izražen, kako se povećava greška očitavanja naviše, objašnjava se time da se koeficijent viška vazduha približava vrednosti koja će dati maksimalnu snagu u odabranoj baznoj radnoj tački. Posle toga, smeša postaje prebogata, što bi dovelo do pada snage i broja obrtaja, ali takve neispravnosti ovde nisu razmatrane.

Promene temperature u usisnom kolektoru nastaju zbog promena masenog EGR protoka, jer je temperatura recirkulisanih izduvni gasova viša od temperature u usisnom kolektoru. Na *Sl. 5.28* vidi se da maseni EGR protok raste, kad se ide od neispravnosti MAF 90% ka neispravnosti MAF 110%. Ovo je posledica doziranja EGR protoka proporcionalno signalu sa MAF senzora, što je ranije izneta pretpostavka. To znači da niža očitana vrednost masenog protoka vazduha u odnosu na stvarnu, dovodi do smanjenja EGR protoka i obrnuto, viša vrednost povećava EGR protok. Imajući sve ovo u vidu, postaje jasno zbog čega je temperatura u usisnom kolektoru pri neispravnostima MAF 90% i MAF 95% niža, a pri neispravnostima MAF 105% i MAF 110% viša od odgovarajuće temperature u stanju bez neispravnosti.

**Neispravnost: Ulaz „falš“-vazduha u usisni sistem motora (FaV 2mm, FaV 4mm, FaV 6mm)**

Zapažanja: Kako pokazuju *Sl. 5.24 - Sl. 5.27*, broj obrtaja motora, maseni protok vazduha pored leptira i temperatura u usisnom kolektoru padaju, dok pritisak u usisnom kolektoru raste. Ove promene su utoliko izraženije, ukoliko je veći otvor kroz koji struji „falš“-

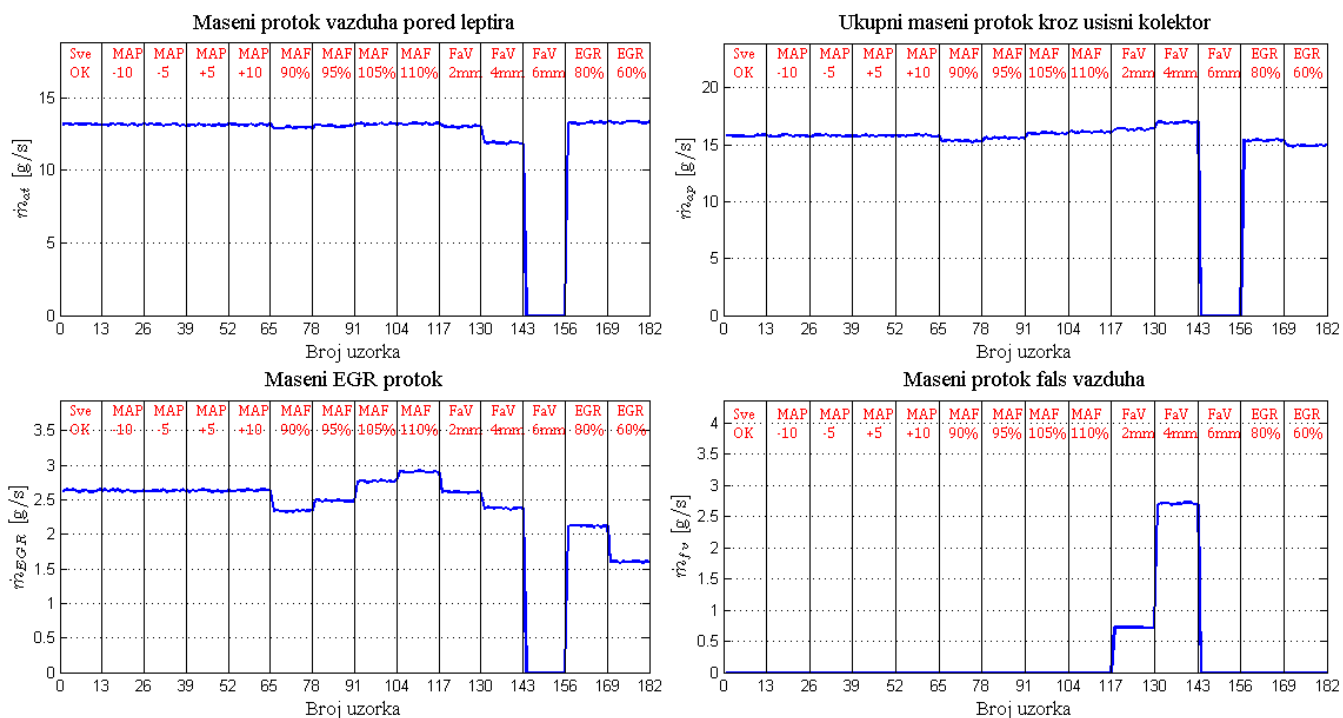
vazduh. Međutim, ono što odmah pada u oči jeste uticaj neispravnosti „FaV 6mm“ na posmatrane veličine. Broj obrtaja i protok pored leptira imaju vrednost nula, što znači da motor ne može da radi sa ovom neispravnošću u odabranoj baznoj radnoj tački. Prirodno, pritisak i temperatura u usisnom kolektoru imaju vrednosti jednake pritisku i temperaturi okoline, jer motor ne radi.

**Objašnjenja:** Ulazak „falš“-vazduha u usisni sistem kod ove vrste motora ostaje „neprimećen“ od strane ECU, jer taj vazduh ne prolazi kroz protokomer. Na taj način, „falš“-vazduh osiromašuje smešu (Sl. 5.23), a s obzirom da u ovom slučaju nema povratne sprege sa  $\lambda$ -sondom, izostaće neophodna korekcija doze goriva. Ovako osiromašena smeša uzrokuje pad snage motora, a time i broja obrtaja.

Pogledom na Sl. 5.28, može se ustanoviti da, zbog dotoka „falš“-vazduha u usisni kolektor, dolazi do pada dotoka vazduha regularnim putem (pored leptira), kao i dotoka recirkulisanih izduvni gasova. Ipak, odnos svih ovih protoka je takav, da je ukupni protok kroz usisni kolektor veći od njegove vrednosti pri radu ispravnog motora. Uglavnom iz ovog razloga, ali i zbog pada broja obrtaja, pritisak u usisnom kolektoru je viši, nego pri radu ispravnog motora.

Pad temperature u usisnom kolektoru tumači se najpre direktnim rashlađujućim efektom koji „falš“-vazduh, kao hladniji, ima na sredinu u usisnom kolektoru. Drugo, „falš“-vazduh utiče na smanjenje EGR protoka, pa na taj način i indirektno doprinosi sniženju temperature u usisnom kolektoru.

Što se tiče neispravnosti „FaV 6mm“, osiromašenje smeše je toliko (Sl. 5.23), da se u motoru ne razvija snaga dovoljna za održavanje stabilnog broja obrtaja ili čak ni ne dolazi do njenog upaljenja.



Sl. 5.28 Relevantni maseni protoci pri simuliranju neispravnosti kod motora sa masenim protokomerom vazduha– Slučaj 1.3

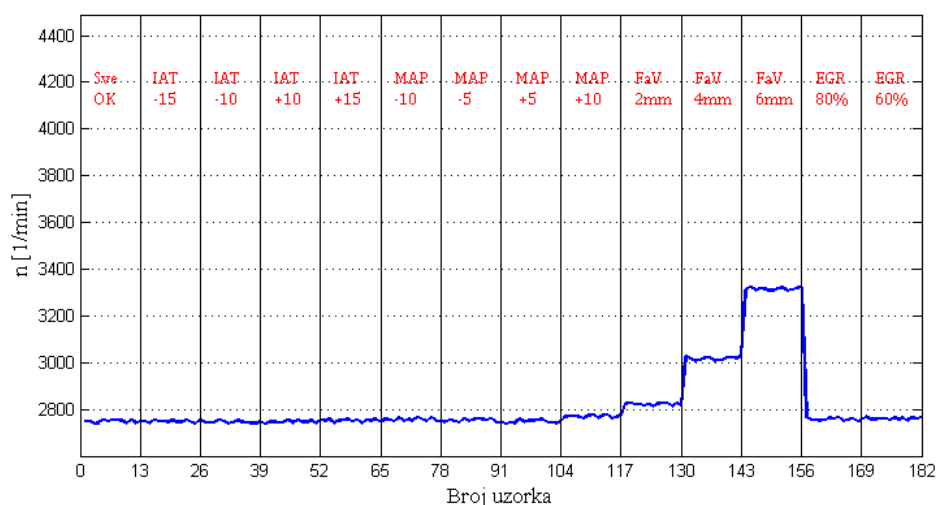
## 5.5.2 Motor bez protokomera vazduha

S obzirom da se kod motora bez protokomera maseni protok kroz usisni kolektor ne meri, već izračunava, skup posmatranih neispravnosti će se nešto razlikovati u odnosu na motore sa protokomerom. Takođe, uticaji pojedinih neispravnosti na posmatrane promenljive, manifestovaće se drugačije, nego kod motora sa protokomerom. Zbog toga se ovde posebno daje pregled rezultata simulacija za sva tri slučaja, opisana u odeljku 5.5.1.

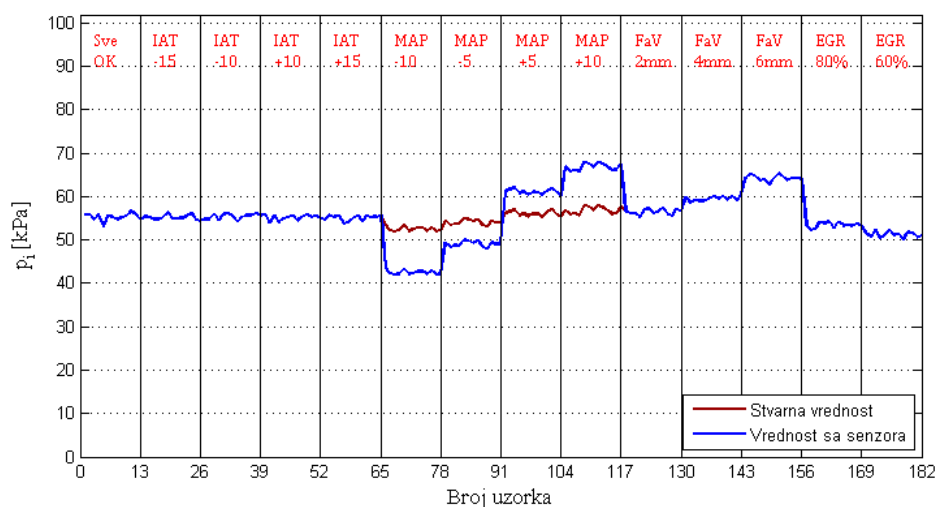
### Slučaj 2.1: Rad u zatvorenoj sprezi sa $\lambda$ -sondom, korekcija vrednosti $\lambda$ nije ograničena

Rezultati simulacija neispravnosti u usisnom sistemu motora bez protokomera, za slučaj kada ne postoji ograničenje korekcije koeficijenta viška vazduha, prikazani su na Sl. 5.29 - Sl. 5.31.

Na osnovu ovih rezultata, mogu se izneti određena zapažanja i komentari po pitanju promene posmatranih karakterističnih veličina motora uzrokovanih uticajem pojedinih neispravnosti, što će biti dato u nastavku.

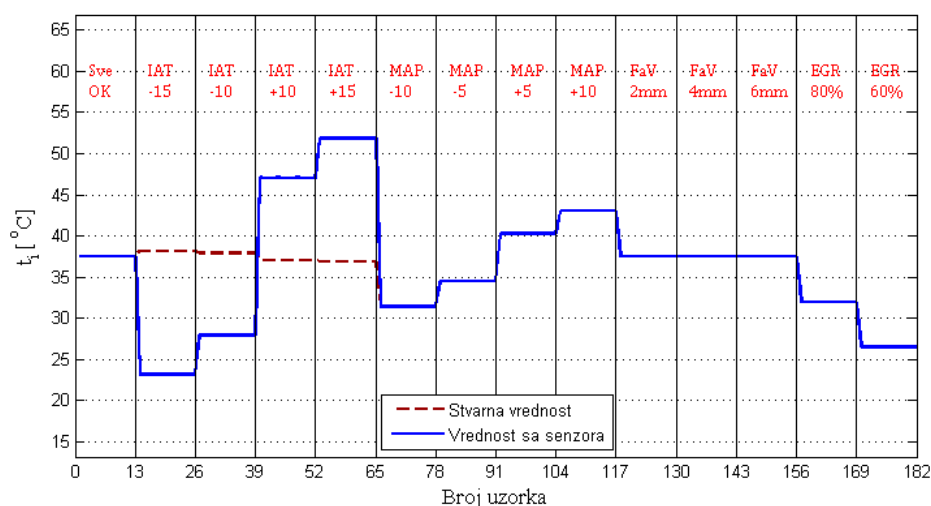


Sl. 5.29 Promena broja obrtaja motora pri simulacijama neispravnosti kod motora bez protokomera vazduha – Slučaj 2.1



Sl. 5.30 Promena pritiska u usisnom kolektoru pri simulacijama neispravnosti kod motora bez protokomera vazduha – Slučaj 2.1





Sl. 5.31 Promena temperature u usisnom kolektoru pri simulacijama neispravnosti kod motora bez protokomera vazduha – Slučaj 2.1

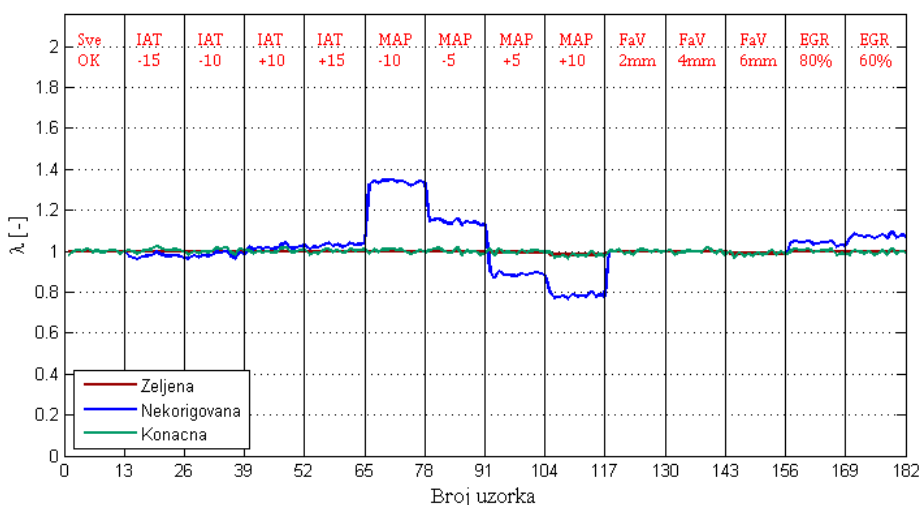
### Neispravnost: Pogrešno očitavanje senzora temperature u usisnom kolektoru (IAT -15, IAT -10, IAT +10, IAT +15)

**Zapažanja:** Vrednosti broja obrtaja (Sl. 5.29) i pritiska u usisnom kolektoru (Sl. 5.30) ostaju na nivou njihovih vrednosti kod ispravnog motora. Jedino što se razlikuje u odnosu na ispravan motor je vrednost sa senzora temperature u usisnom kolektoru, kako se vidi na Sl. 5.31. Ova vrednost je manja ili veća za 15 ili 10 stepeni od stvarne vrednosti, u zavisnosti od konkretne neispravnosti. Na istoj slici se može primetiti i da stvarna vrednost temperature u usisnom kolektoru blago odstupa od njene vrednosti kod ispravnog motora i to naviše pri greškama „IAT -15“ i „IAT -10“, a naniže pri greškama „IAT +10“ i „IAT +15“.

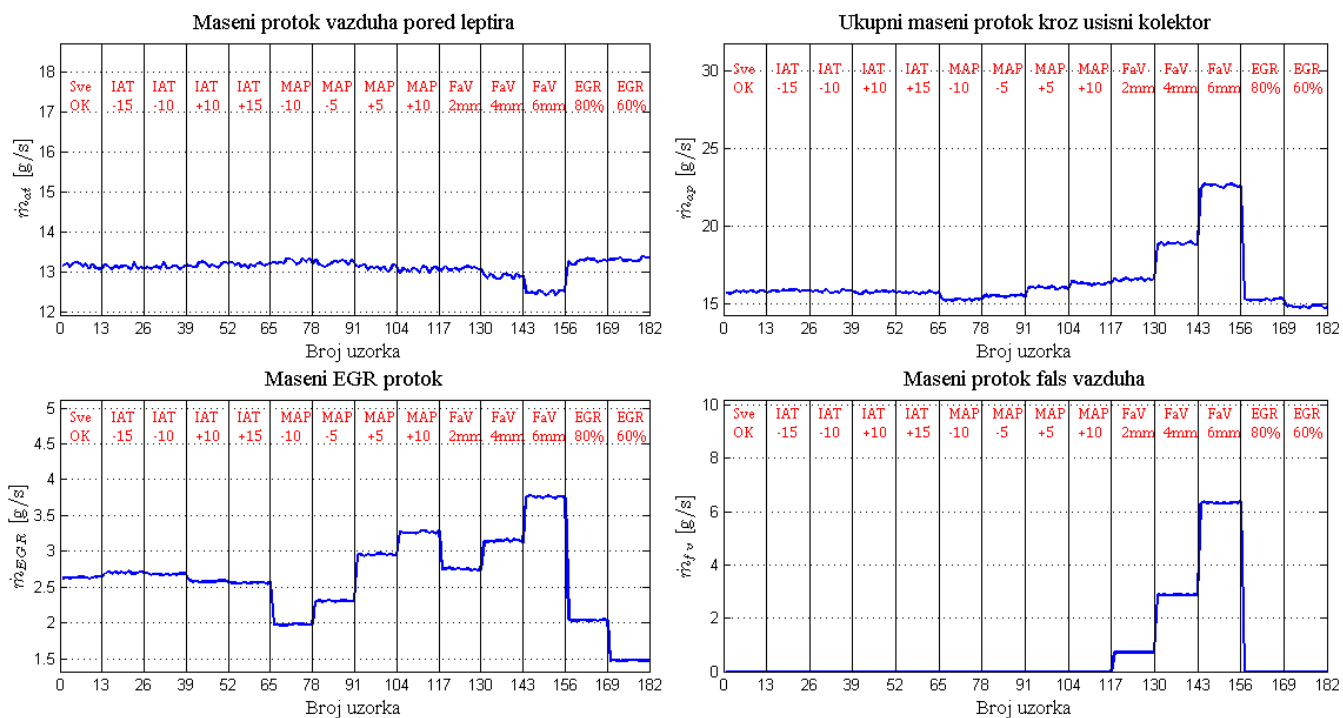
**Objašnjenja:** Kod ove vrste motora, senzor temperature u usisnom kolektoru ima uticaja na doziranje goriva, jer gustina vazduha, a time i njegov izračunati maseni protok zavise od temperature. Međutim, za pogrešna očitavanja reda veličine 10-15 stepeni, greška u računanju masenog protoka je relativno mala. Ovo je ilustrovano na Sl. 5.32, koja pokazuje da je odstupanje nekorigovane vrednosti koeficijenta viška vazduha od njegove željene vrednosti jedva primetno. Čak i u slučaju većeg odstupanja, ne bi došlo do promene snage motora, a time ni njegovog broja obrtaja, zbog rada motora u zatvorenoj sprezi sa  $\lambda$ -sondom. Drugim rečima, dolazi do odgovarajuće korekcije doze goriva, čime se neutrališe greška očitavanja senzora temperature u usisnom kolektoru.

Promena stvarne vrednosti temperature u usisnom kolektoru je posledica promene masenog EGR protoka, jer je temperatura recirkulisanih izduvni gasova viša od temperature u usisnom kolektoru. Na Sl. 5.33 vidi se da maseni EGR protok blago opada, kad se ide od neispravnosti „IAT -15“ ka neispravnosti „IAT +15“. Maseni EGR protok ima ovakvu promenu jer je u simulatoru neispravnosti pretpostavljeno da je doziranje EGR protoka proporcionalno veličini masenog protoka vazduha kroz usisni kolektor, koju određuje ECU. Jedan od senzora koji omogućavaju ECU da odredi maseni protok vazduha, kod motora bez protokomera, je i senzor temperature u usisnom kolektoru. Očitavanjem niže vrednosti temperature u usisnom kolektoru, ECU izračunava veću vrednost masenog protoka vazduha i obrnuto, zbog očitavanja više vrednosti temperature, ECU će izračunati vrednost masenog protoka vazduha koja je manja od stvarnog protoka. Na osnovu ove diskusije, sledi da greške „IAT -15“ i „IAT -10“ dovode do blagog porasta masenog EGR protoka, a time i stvarne

temperature u usisnom kolektoru, dok greške „IAT +10“ i „IAT +15“ blago umanjuju EGR protok i stvarnu temperaturu u usisnom kolektoru.



Sl. 5.32 Karakteristične vrednosti koeficijenta viška vazduha pri simuliranju neispravnosti kod motora bez protokomera vazduha – Slučaj 2.1



Sl. 5.33 Relevantni maseni protoci pri simuliranju neispravnosti kod motora bez protokomera vazduha – Slučaj 2.1

**Neispravnost: Pogrešno očitavanje senzora pritiska u usisnom kolektoru (MAP -10, MAP -5, MAP +5, MAP +10)**

**Zapažanja:** Ova neispravnost ima osetnijeg uticaja na pritisak i temperaturu u usisnom kolektoru, a broj obrtaja ostaje na nivou koji motor ima pri radu bez neispravnosti.

Očitana vrednost senzora pritiska u usisnom kolektoru je manja ili veća za 10 ili 5 kPa od stvarne vrednosti, u zavisnosti od konkretne neispravnosti (Sl. 5.30). Na Sl. 5.30 može se primetiti i da stvarna vrednost pritiska u usisnom kolektoru blago odstupa od njene vrednosti kod ispravnog motora i to naniže pri greškama „MAP -10“ i „MAP -5“, a naviše pri greškama „MAP +5“ i „MAP +10“.

Pri postojanju grešaka „MAP -10“ i „MAP -5“, nivo temperature u usisnom kolektoru je niži u odnosu na isti u stanju bez neispravnosti, što se vidi na *Sl. 5.31*. Obrnuto, greške „MAP +5“ i „MAP +10“ imaju za posledicu višu temperaturu u usisnom kolektoru u odnosu na njen nivo pri ispravnom stanju.

**Objašnjenja:** Kod ove vrste motora, očekuje se da pogrešno očitavanje davača pritiska u usisnom kolektoru ima uticaja na rad motora, jer je njegov signal veoma bitan za doziranje goriva. Na *Sl. 5.32* vidi se kakav je uticaj ove neispravnosti na vrednost koeficijenta viška vazduha. Primećuje se da greške „MAP -10“ i „MAP -5“ dovode do porasta koeficijenta viška vazduha u odnosu na željenu vrednost. Ovo je logično, jer ECU izračunava nižu vrednost masenog protoka vazduha kroz usisni kolektor, pa smanjuje dozu goriva, a zapravo se protok vazduha nije promenio. Obrnuta je situacija kod grešaka „MAP +5“ i „MAP +10“, koje dovode do obogaćenja smeše. Međutim, pošto u ovom slučaju motor radi u režimu zatvorene sprege sa  $\lambda$ -sondom, dolazi do odgovarajuće korekcije doze goriva kojom se konačna vrednost koeficijenta viška vazduha dovodi na željeni nivo. Drugim rečima, zatvorena sprega sa  $\lambda$ -sondom praktično ispravlja grešku očitavanja senzora pritiska. Otuda motor dobija smešu, koja po količini i sastavu odgovara željenoj smeši. Stoga broj obrtaja ostaje na nivou onog koji ima ispravan motor.

Promene stvarne vrednosti pritiska u usisnom kolektoru mogu se objasniti pomoću *Sl. 5.33*. Tamo se vidi da ukupni maseni protok kroz usisni kolektor blago raste, posmatrano od greške „MAP -10“ ka grešci „MAP +10“. Pošto je broj obrtaja nepromenjen, i stvarna vrednost pritiska u usisnom kolektoru će u posmatranoj zoni dijagrama blago da raste (*Sl. 5.30*).

Na *Sl. 5.33* vidi se da greške „MAP -10“ i „MAP -5“ umanjuju maseni EGR protok, a greške „MAP +5“ i „MAP +10“ ga uvećavaju. Razlog ovakvog ponašanja EGR protoka je taj, što ECU usklađuje maseni EGR protok sa masenim protokom vazduha kroz usisni kolektor, kako je ranije već objašnjeno. Umanjeni EGR protok, zbog nepromenjenog broja obrtaja, izaziva blago povećanje dotoka vazduha pored leptira, što, sve zajedno, dovodi do pada temperature u usisnom kolektoru.

Suprotno, uvećani EGR protok blago umanjuje dotok vazduha pored leptira, takođe zbog nepromenjenog broja obrtaja. Ovo dovodi do porasta temperature u usisnom kolektoru u odnosu na njenu vrednost kod ispravnog motora.

### **Neispravnost: Ulaz „falš“-vazduha u usisni sistem motora (FaV 2mm, FaV 4mm, FaV 6mm)**

**Zapažanja:** Većina posmatranih veličina se menja pod uticajem ove neispravnosti. Broj obrtaja motora (*Sl. 5.29*) i pritisak u usisnom kolektoru (*Sl. 5.30*) se povećavaju, i to sve više, što je veći otvor kroz koji struji „falš“-vazduh. Za razliku od njih, temperatura u usisnom kolektoru zadržava vrednost koju ima pri radu motora bez neispravnosti (*Sl. 5.31*).

**Objašnjenja:** Kod motora bez protokomera, ova neispravnost dovodi do promena sličnih onima, koje bi nastale pri odgovarajućem povećanju otvora leptira. To je zato što, kod ovih motora, ECU izračunava maseni protok vazduha kroz usisni kolektor na osnovu signala senzora pritiska i temperature u usisnom kolektoru i senzora broja obrtaja motora. Pošto je pretpostavka da u isto vreme ne može da postoji više neispravnosti, svi pomenuti senzori su ispravni, tako da ECU izračunava tačnu vrednost masenog protoka kroz usisni kolektor, bez obzira što je taj protok jednim delom ostvaren neželjenim putem. Sve u svemu, tačno određen maseni protok vazduha kroz usisni kolektor, dovodi i do tačnog određivanja doze goriva. Zbog toga će smeša koja odlazi u motor biti željenog sastava, bez potrebe za bilo kakvom korekcijom u povratnoj sprezi sa  $\lambda$ -sondom, što se vidi na *Sl. 5.32*. Jedino će količina te smeše biti veća od

količine uobičajene za rad ispravnog motora u posmatranoj baznoj radnoj tački. Zato se povećava snaga motora, a time i njegov broj obrtaja.

Promena pritiska u usisnom kolektoru može se objasniti pomoću *Sl. 5.33*. Ova slika pokazuje da, bez obzira što pojava „falš“-vazduha umanjuje protok vazduha pored leptira, ukupan protok kroz usisni kolektor raste. Pozitivan uticaj porasta ovog protoka očigledno je dominantniji od negativnog uticaja porasta broja obrtaja na pritisak u usisnom kolektoru, tako da je rezultat - povećanje pritiska u usisnom kolektoru.

Temperatura u usisnom kolektoru se ne menja, bez obzira na dolazak hladnijeg „falš“-vazduha. Tumačenje ove situacije ponovo se zasniva na *Sl. 5.33*., koja pokazuje da se i EGR protok povećava sa porastom protoka „falš“-vazduha. Ovo je u skladu sa ranije izrečenom pretpostavkom o doziranju masenog EGR protoka proporcionalno izračunatom masenom protoku vazduha kroz usisni kolektor. Zbog ove proporcionalnosti, održava se nepromenjena temperatura u usisnom kolektoru.

### **Neispravnost: Umanjen EGR protok (EGR 80%, EGR 60%)**

Zapažanja: Ova neispravnost ima primetan uticaj na pritisak i temperaturu u usisnom kolektoru, a zanemariv uticaj na broj obrtaja motora. Kako se vidi na *Sl. 5.30* i *Sl. 5.31*, pritisak i temperatura u usisnom kolektoru su niži u odnosu na njihove vrednosti kod ispravnog motora. Što je veće umanjenje EGR protoka, ove posledice su izraženije. Broj obrtaja motora se neznatno povećava sa umanjenjem masenog EGR protoka.

Objašnjenja: Razlog opadanja pritiska u usisnom kolektoru, u odnosu na stanje bez neispravnosti, može se pronaći na *Sl. 5.33*. Tamo se vidi da je ukupni maseni protok kroz usisni kolektor manji, nego kod ispravnog motora, bez obzira što usled umanjenja EGR protoka dolazi do blagog porasta dotoka vazduha pored leptira. Pad ukupnog masenog protoka kroz usisni kolektor, uz činjenicu da broj obrtaja motora ostaje gotovo nepromenjen, potpuno objašnjava niži pritisak u usisnom kolektoru pri postojanju razmatrane neispravnosti.

Do pada temperature dolazi zbog manjeg dotoka recirkulisanih izduvni gasova, čija temperatura je znatno viša od temperature u usisnom kolektoru. Osim toga, blago raste dotok vazduha u usisni kolektor pored leptira (*Sl. 5.33*). Otuda će zagrevanje sredine u usisnom kolektoru biti slabije.

Vrlo blago povećanje masenog protoka pored leptira dovodi do odgovarajućeg povećanja doze goriva, što sve zajedno daje veoma, veoma blago povećanje snage motora. Otuda i neznatno veći broj obrtaja u odnosu na isti kod ispravnog motora.

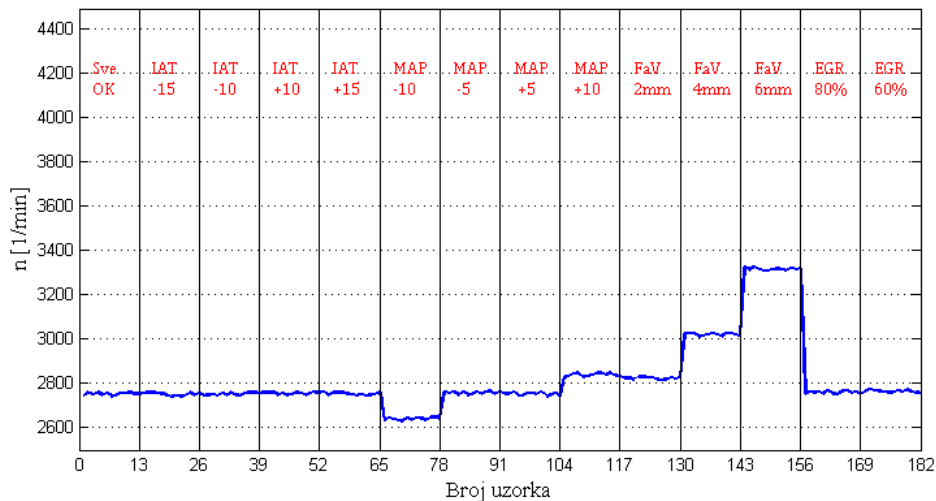
### **Slučaj 2.2: Rad u zatvorenoj sprezi sa $\lambda$ -sondom, korekcija vrednosti $\lambda$ ograničena**

Kao kod Slučaja 2.1, i ovde je u pitanju rad motora u zatvorenoj sprezi sa lambda-sondom, pa među njima ima veoma malo razlika. Zbog ograničenja veličine korekcije pri doziranju goriva, eventualne razlike mogu da se pojave samo kod neispravnosti koje dovode do većeg poremećaja koeficijenta viška vazduha. Tada koeficijent viška vazduha ne može da se dovede na nivo željene vrednosti, kako je objašnjeno u Odeljku 5.3.7.3. Za maksimalnu dozvoljenu vrednost korekcije u ovom istraživanju je usvojena vrednost 20%.

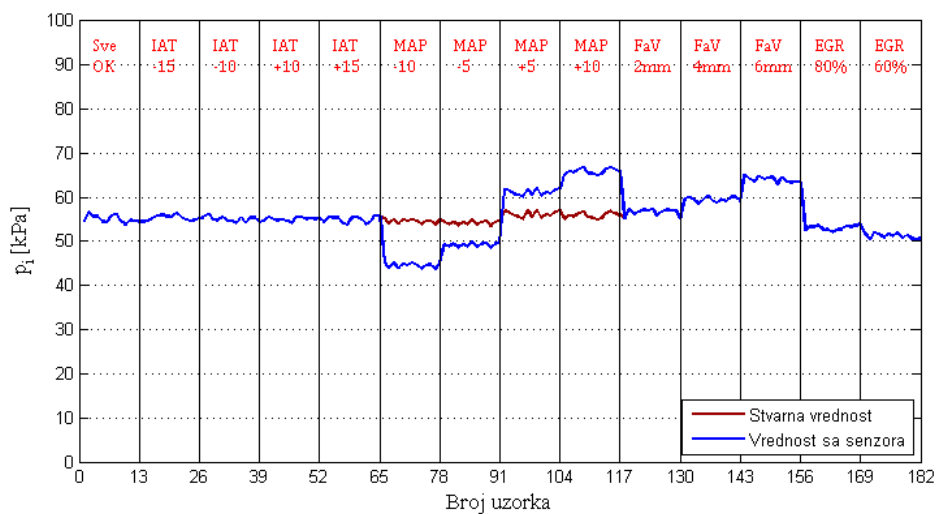
U nastavku će biti prikazani rezultati simulacija za ovaj slučaj, a u diskusiji će se pažnja posvetiti samo onim neispravnostima, koje ga čine drugačijim od Slučaja 2.1. Kod svih ostalih neispravnosti, komentari bi bili potpuno isti, kao u Slučaju 2.1, pa su ovde izostavljeni, da bi se izbeglo nepotrebno ponavljanje.

Uticaji pojedinih simuliranih neispravnosti na posmatrane promenljive motora prikazani su na *Sl. 5.34* - *Sl. 5.36*. Poređenjem ovih slika sa *Sl. 5.29* - *Sl. 5.31*, respektivno, primećuje se da se, za posmatrane intenzitete neispravnosti, Slučajevi 2.2 i 2.1 razlikuju samo po neispravnostima „MAP

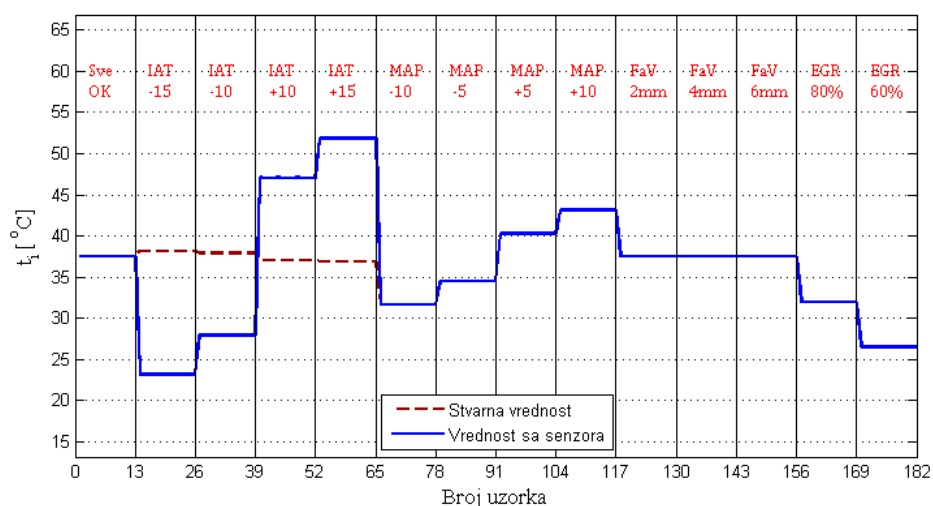
-10" i „MAP +10“. Zbog toga će ovde biti prodiskutovane samo te dve neispravnosti. Pri radu motora u nekim drugim baznim radnim tačkama i pri simuliranju nekih drugih intenziteta neispravnosti, verovatno bi se pojavile još neke razlike između Slučaja 2.2 i Slučaja 2.1, ali ovde nije cilj da se pronađu sve te razlike, nego da se jednim primerom ilustruje situacija u kojoj ne može da se dostigne željena vrednost koeficijenta viška vazduha.



Sl. 5.34 Promena broja obrtaja motora pri simulaciji neispravnosti kod motora bez protokomera vazduha – Slučaj 2.2



Sl. 5.35 Promena pritiska u usisnom kolektoru pri simulacijama neispravnosti kod motora bez protokomera vazduha – Slučaj 2.2



Sl. 5.36 Promena temperature u usisnom kolektoru pri simulacijama neispravnosti kod motora bez protokomera vazduha – Slučaj 2.2

### Neispravnost: Pogrešno očitavanje senzora pritiska u usisnom kolektoru (MAP -10, MAP +10)

**Zapažanja:** Pojava neispravnosti „MAP -10“ dovodi do pada broja obrtaja u odnosu na njegovu vrednost kod ispravnog motora (Sl. 5.34). Očitavanje senzora pritiska u usisnom kolektoru je za 10 kPa manje od stvarne vrednosti, koja ostaje gotovo nepromenjena u odnosu na ispravan motor (Sl. 5.35). Temperatura u usisnom kolektoru je niža nego u stanju bez neispravnosti (Sl. 5.36).

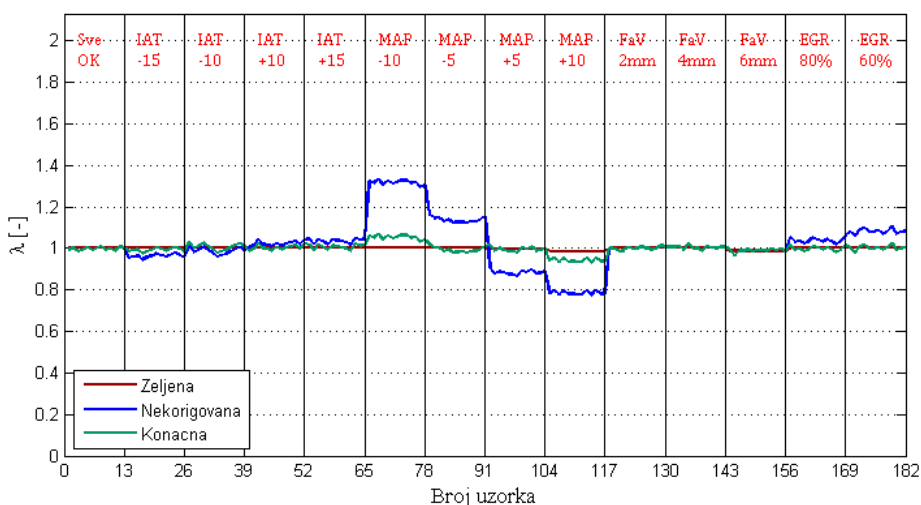
Greška „MAP +10“ ima suprotne posledice od greške „MAP -10“. Naime, dolazi do uvećanja broja obrtaja i temperature u usisnom kolektoru, u poređenju sa stanjem bez neispravnosti. Očitana vrednost senzora pritiska je za 10 kPa veća od stvarne vrednosti pritiska u usisnom kolektoru, koja ostaje približno na nivou vrednosti kod ispravnog motora.

**Objašnjenja:** Greška „MAP -10“ uzrokuje izračunavanje niže vrednosti masenog protoka vazduha kroz usisni kolektor, nego što ona zaista jeste. To dovodi do smanjenje doze goriva, što za rezultat ima osiromašenje smeše, kako pokazuje Sl. 5.37. Međutim, pošto je u funkciji povratna sprega sa  $\lambda$ -sondom, vrši se korekcija doze goriva. S obzirom da je u ovom slučaju korekcija ograničena na 20%, a odstupanje nekorigovane u odnosu na željenu vrednost  $\lambda$  je relativno veliko, nije moguće dovesti sastav smeše na željeni nivo. I pored maksimalne korekcije, smeša ostaje blago siromašna, što u negativnom smislu utiče na broj obrtaja motora.

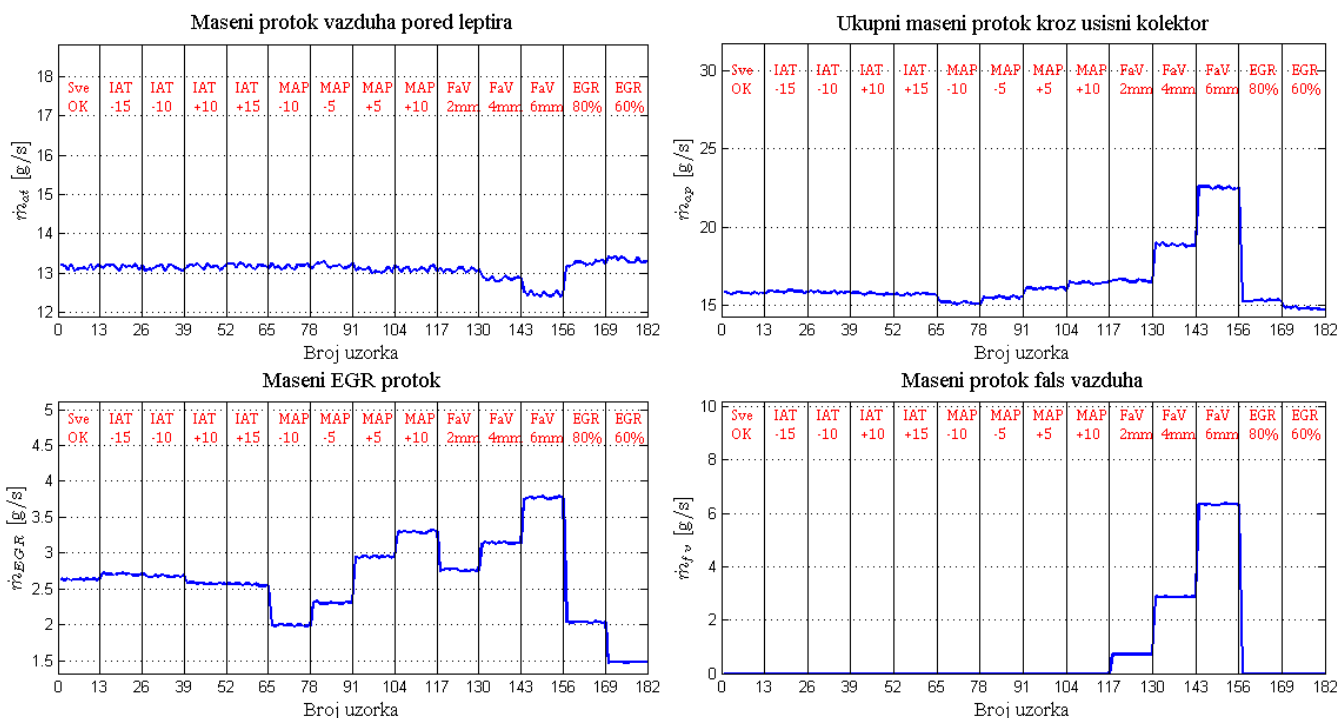
Nepromenjena vrednost pritiska u usisnom kolektoru u prisustvu greške „MAP -10“ nije baš logična, kada se zna da je došlo do smanjenja broja obrtaja, jer pad broja obrtaja uglavnom dovodi do porasta pritiska u usisnom kolektoru. Ova „zagonetka“ se može razrešiti, ako se pogleda Sl. 5.38, gde se vidi da je i ukupni maseni protok kroz usisni kolektor umanjen. Imajući i ovo u vidu, jasno je zašto pritisak ostaje na nivou koji ima pri radu ispravnog motora. Umanjenje vrednosti temperature u usisnom kolektoru direktna je posledica smanjenog EGR protoka, što se takođe vidi na Sl. 5.38, a detalji su već objašnjeni u Slučaju 2.1.

Što se tiče posledica greške „MAP +10“, one se mogu objasniti na potpuno isti način, kao kod greške „MAP - 10“. Zbog toga će to biti urađeno sa manje detalja. Dakle, greška „MAP +10“ dovodi do obogaćenja smeše (Sl. 5.37), koja i pored korekcije ostaje u oblasti  $\lambda < 1$ , pa broj obrtaja raste. Porast broja obrtaja ne dovodi do pada pritiska u usisnom kolektoru, jer se i ukupan maseni protok kroz usisni kolektor

povećava. Viša temperatura u usisnom kolektoru uzrokovana je povećanjem masenog EGR protoka, što pokazuje Sl. 5.38.



Sl. 5.37 Karakteristične vrednosti koeficijenta viška vazduha pri simuliranju neispravnosti kod motora bez protokomera vazduha – Slučaj 2.2

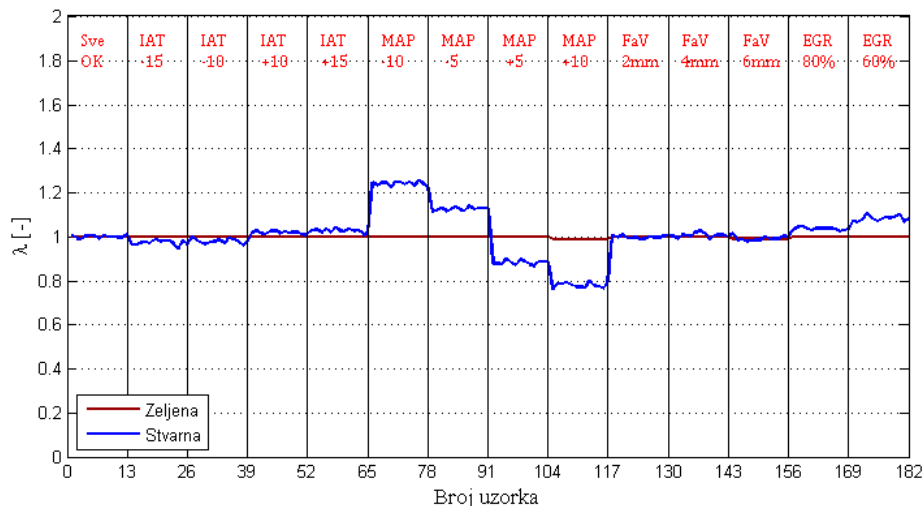


Sl. 5.38 Relevantni maseni protoci pri simuliranju neispravnosti kod motora bez protokomera vazduha – Slučaj 2.2

### Slučaj 2.3: Rad u otvorenoj sprezi sa $\lambda$ -sondom

Simulacijama neispravnosti u usisnom sistemu motora bez protokomera, koji radi u režimu otvorene sprege sa lambda-sondom, dobijeni su rezultati, prikazani na Sl. 5.39 - Sl. 5.43.

Zbog odsustva povratne sprege sa  $\lambda$ -sondom, posebnu pažnju treba posvetiti neispravnostima koje dovode do odstupanja koeficijenta viška vazduha od željene vrednosti. Pošto korekcija vrednosti koeficijenta viška vazduha nije moguća, ove neispravnosti dovode do promena vrednosti nekih posmatranih promenljivih u odnosu na ispravno stanje motora. Koje su to neispravnosti, najbolje se može videti sa dijagrama koeficijenta viška vazduha, prikazanog na Sl. 5.39.



Sl. 5.39 Karakteristične vrednosti koeficijenta viška vazduha pri simuliranju neispravnosti kod motora bez protokomera vazduha – Slučaj 2.3

Kako pokazuje Sl. 5.39, neispravnost „ulaz falš-vazduha u usisni sistem“ („FaV 2mm“, „FaV 4mm“ i „FaV 6mm“) ne dovodi do poremećaja koeficijenta viška vazduha, pa bi zapažanja i objašnjenja, koja se odnose na ovu neispravnost, bila identična onima u Slučaju 2.1. Da bi se izbeglo ponavljanje, komentari o ovim neispravnostima će ovde biti izostavljeni, a pažnja će biti posvećena preostalim neispravnostima.

Na Sl. 5.40 - Sl. 5.42 prikazane su promene posmatranih karakterističnih promenljivih usled simuliranih neispravnosti, a kakve su to promene, biće prodiskutovano u nastavku.

### Neispravnost: Pogrešno očitavanje senzora temperature u usisnom kolektoru (IAT -15, IAT -10, IAT +10, IAT +15)

**Zapažanja:** Greške „IAT -15“ i „IAT -10“ imaju za posledicu povećanje broja obrtaja, dok suprotno važi za greške „IAT +10“ i „IAT +15“ (Sl. 5.40). Što je greška veća, i promena broja obrtaja je veća.

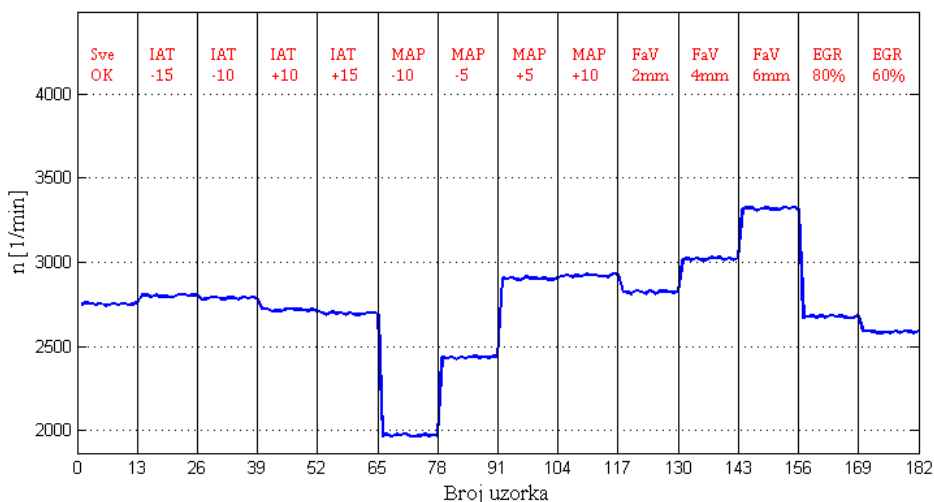
Očitana vrednost sa senzora temperature u usisnom kolektoru je manja ili veća za 15 ili 10 stepeni od stvarne vrednosti, u zavisnosti od konkretne neispravnosti, što je ilustrovano na Sl. 5.42. Na istoj slici se uočava i da stvarna vrednost temperature u usisnom kolektoru blago odstupa od njene vrednosti kod ispravnog motora i to naviše pri greškama „IAT -15“ i „IAT -10“, a naniže pri greškama „IAT +10“ i „IAT +15“.

Na pritisku u usisnom kolektoru ne primećuju se značajnije promene u odnosu na njegovu vrednost kod ispravnog motora.

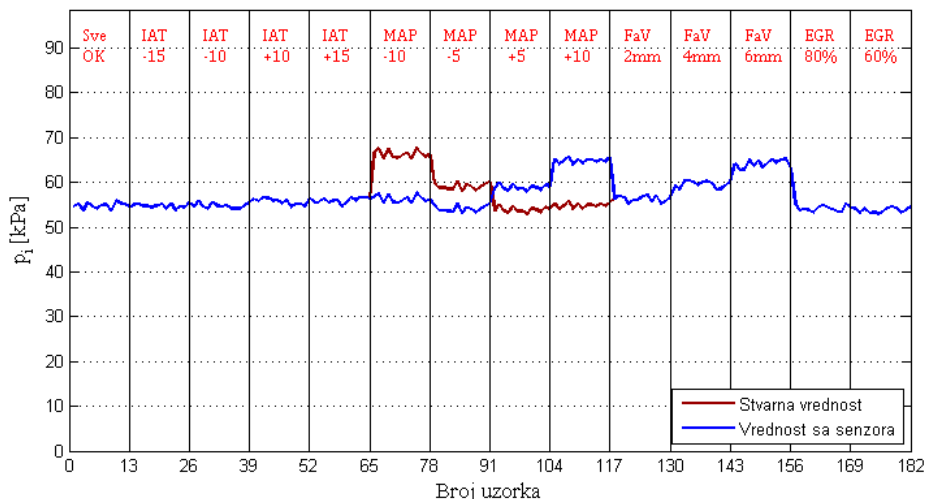
**Objašnjenja:** Kod ove vrste motora, maseni protok vazduha kroz usisni kolektor izračunava se na osnovu signala sa nekoliko senzora, među kojima je i senzor temperature u usisnom kolektoru. Pri greškama očitavanja naniže u odnosu na stvarnu vrednost temperature, ECU će izračunati veći maseni protok vazduha od stvarnog, što će dovesti do povećanja doze goriva i obogaćenja smeše (Sl. 5.39). Suprotno tome, ako senzor temperature pogrešno očitava naviše, smeša će osiromašiti. Međutim, za pogrešna očitavanja reda veličine 10-15 stepeni, greška u računanju masenog protoka je relativno mala, pa je odstupanje stvarne vrednosti koeficijenta viška vazduha od njegove željene vrednosti jedva primetno. Pošto nema povratne sprege sa  $\lambda$ -sondom, i ova mala odstupanja će dovesti do određenih, isto tako malih, promena u razvijenoj snazi motora, a time i broju obrtaja. Tako će greške „IAT -15“ i „IAT -10“ veoma blago povećati broj obrtaja, a greške „IAT +10“ i „IAT +15“ će ga umanjiti.



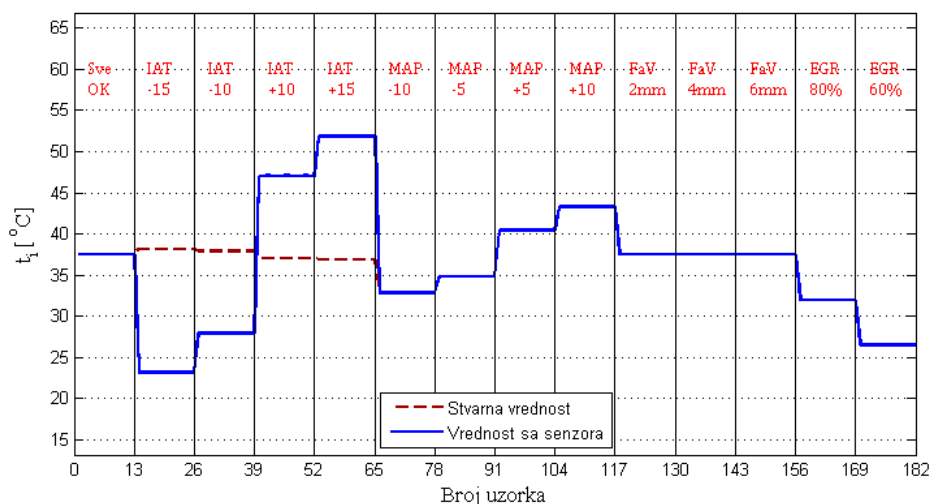
Promena stvarne vrednosti temperature u usisnom kolektoru je posledica promene masenog EGR protoka, jer je temperatura recirkulisanih izduvni gasova viša od temperature u usisnom kolektoru. Na Sl. 5.43 vidi se da maseni EGR protok blago opada, kad se ide od neispravnosti „IAT -15“ ka neispravnosti „IAT +15“. Maseni EGR protok ima ovakvu promenu jer je u simulatoru neispravnosti pretpostavljeno da je doziranje EGR protoka proporcionalno veličini izračunatog masenog protoka vazduha kroz usisni kolektor. Očitavanjem niže vrednosti temperature u usisnom kolektoru, ECU izračunava veću vrednost masenog protoka vazduha i obrnuto. Na osnovu ove diskusije, može se zaključiti da greške „IAT -15“ i „IAT -10“ dovode do blagog porasta masenog EGR protoka, a time i porasta stvarne temperature u usisnom kolektoru, dok greške „IAT +10“ i „IAT +15“ blago smanjuju EGR protok i stvarnu temperaturu u usisnom kolektoru.



Sl. 5.40 Promena broja obrtaja motora pri simulacijama neispravnosti kod motora bez protokomera vazduha – Slučaj 2.3



Sl. 5.41 Promena pritiska u usisnom kolektoru pri simulacijama neispravnosti kod motora bez protokomera vazduha – Slučaj 2.3



Sl. 5.42 Promena temperature u usisnom kolektoru pri simulacijama neispravnosti kod motora bez protokomera vazduha – Slučaj 2.3

### Neispravnost: Pogrešno očitavanje senzora pritiska u usisnom kolektoru (MAP -10, MAP -5, MAP +5, MAP +10)

**Zapažanja:** Očitavanje nižeg pritiska u usisnom kolektoru („MAP -10“ i „MAP -5“) u odnosu na njegovu stvarnu vrednost, dovodi do osetnog umanjenja broja obrtaja. Ove posledice su sve izraženije, što je intenzitet neispravnosti veći.

Obrnuto, očitavanje veće vrednosti pritiska u usisnom kolektoru („MAP +5“ i „MAP +10“) dovodi do povećanja broja obrtaja, u poređenju sa stanjem ispravnog motora. Sa Sl. 5.40, može se primetiti da je broj obrtaja kod neispravnosti „MAP +10“ tek vrlo malo veći, nego kod neispravnosti „MAP +5“. Otuda se može pretpostaviti da bi neka još veća greška očitavanja dovela do obrnutog trenda – pada broja obrtaja.

Očitana vrednost senzora pritiska u usisnom kolektoru je manja ili veća za 10 ili 5 kPa od stvarne vrednosti, u zavisnosti od konkretne neispravnosti (Sl. 5.41). Na Sl. 5.41 uočava se i da je stvarna vrednost pritiska u usisnom kolektoru pri greškama „MAP -10“ i „MAP -5“, značajno viša od njene vrednosti kod ispravnog motora, dok greške „MAP +5“ i „MAP +10“ gotovo da nemaju uticaja na stvarnu vrednost pritiska.

U prisustvu grešaka „MAP -10“ i „MAP -5“, temperatura u usisnom kolektoru je niža u odnosu na istu kod ispravnog motora, što se vidi na Sl. 5.42. Obrnuto, greške „MAP +5“ i „MAP +10“ podižu temperaturu u usisnom kolektoru u odnosu na njen nivo u stanju bez neispravnosti.

**Objašnjenja:** Greške „MAP -10“ i „MAP -5“ dovode do smanjenja doze goriva, tako da smeša postaje siromašnija (Sl. 5.39). Zato dolazi do umanjenja snage motora, a time i do osetnog pada broja obrtaja. Usled primetno nižeg broja obrtaja, vrednost pritiska u usisnom kolektoru je viša, nego kod ispravnog motora (Sl. 5.41), bez obzira na umanjeni ukupni maseni protok kroz usisni kolektor (Sl. 5.43).

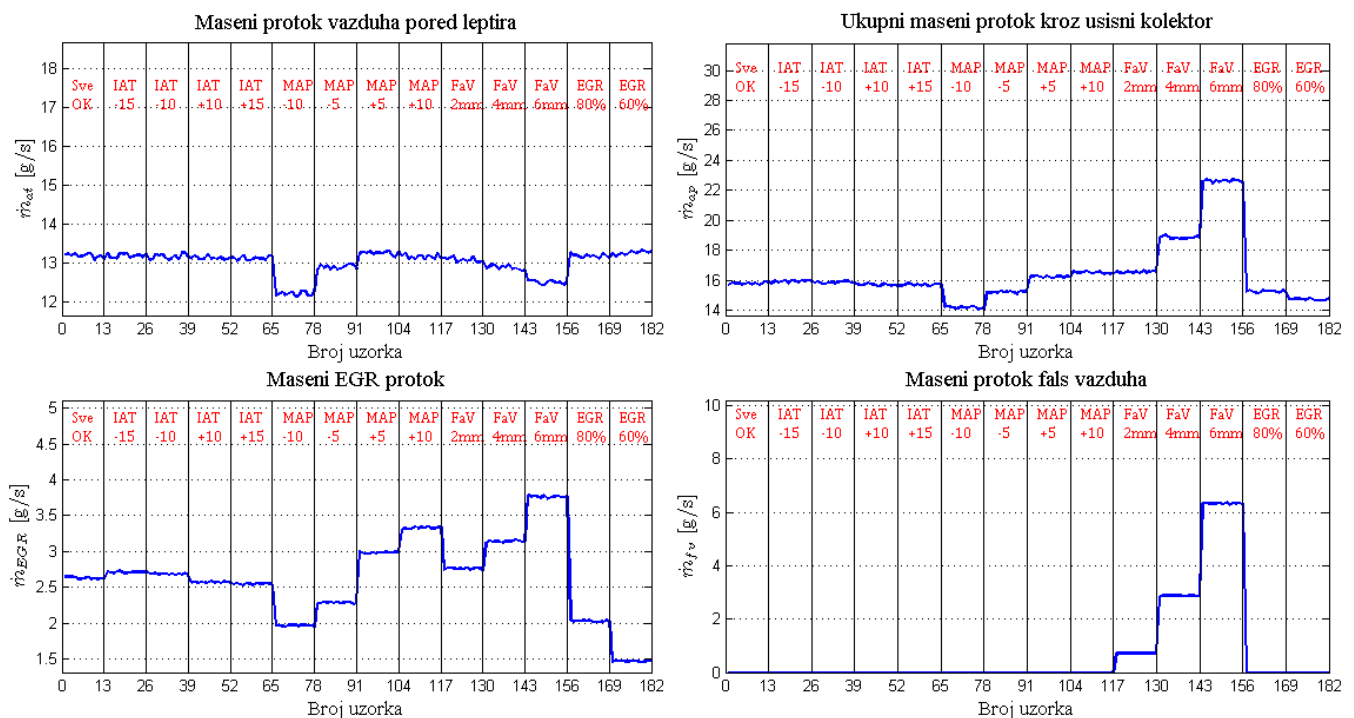
Greške „MAP +5“ i „MAP +10“ uzrokuju povećanje doze goriva, a to znači i obogaćenje smeše, kako pokazuje Sl. 5.39. Otuda je snaga motora nešto veća, nego kod ispravnog motora, pa dolazi do porasta broja obrtaja. Veći broj obrtaja bi trebalo da dovede do pada pritiska u usisnom kolektoru, ali do toga ne dolazi, jer je ukupni maseni protok kroz usisni kolektor povećan (Sl. 5.43).

Činjenica da je porast broja obrtaja sve slabije izražen, kako se povećava greška očitavanja naviše (od „MAP +5“ ka „MAP +10“), objašnjava se time da se koeficijent viška vazduha približava vrednosti koja će dati maksimalnu snagu u odabranoj

baznoj radnoj tački. Posle toga, smeša postaje prebogata, što bi dovelo do pada snage i broja obrtaja.

Na Sl. 5.43 vidi se da greške „MAP -10“ i „MAP -5“ umanjuju maseni EGR protok, a greške „MAP +5“ i „MAP +10“ ga uvećavaju. To je zato što ECU usklađuje maseni EGR protok sa izračunatim masenim protokom vazduha kroz usisni kolektor, kako je ranije već objašnjeno. Usled manjeg EGR protoka slabije je zagrevanje usisnog kolektora, pa temperatura u njemu pada.

Suprotno tome, veći EGR protok pojačava zagrevanje usisnog kolektora, uzrokujući povećanje temperature u odnosu na njenu vrednost kod ispravnog motora.



Sl. 5.43 Relevantni maseni protoci pri simuliranju neispravnosti kod motora bez protokomera vazduha – Slučaj 1.3

### Neispravnost: Umanjen EGR protok (EGR 80%, EGR 60%)

**Zapažanja:** Ova neispravnost ima određenog uticaja na broj obrtaja motora i temperaturu u usisnom kolektoru, dok je uticaj na pritisak u usisnom kolektoru zanemariv.

Kako se vidi na Sl. 5.40, broj obrtaja je niži u odnosu na njegovu vrednost kod ispravnog motora. Što je veće umanjeње EGR protoka, ove posledice su izraženije.

Potpuno isti karakter ima i uticaj ove neispravnosti na temperaturu u usisnom kolektoru. To je ilustrovano na Sl. 5.42, gde se vidi da je temperatura u usisnom kolektoru niža, što je veće zagušenje u EGR vodu.

**Objašnjenja:** Razlog blagog smanjenja broja obrtaja može se videti na Sl. 5.39, koja pokazuje da sa povećanjem zagušenja EGR voda, smeša postaje siromašnija. Takva smeša ima za posledicu umanjeње snage motora, a time i broja obrtaja. Međutim, postavlja se pitanje zašto smeša postaje siromašnija pri zagušenju u EGR vodu. Odgovor nudi Sl. 5.43, na kojoj se vidi da umanjeње maseni EGR protok dovodi do blagog porasta masenog protoka vazduha pored leptira. Maseni protok goriva, izračunat prema jednačini (5.46), nije dovoljan da se sa uvećanim protokom pored leptira formira smeša željenog sastava, tako da ona ostaje blago siromašna.

Do pada temperature dolazi zbog manjeg dotoka recirkulisanih izduvni gasova, čija temperatura je znatno viša od temperature u usisnom kolektoru.

### 5.5.3 Zaključni komentar rezultata simulacija neispravnosti

U odeljcima 5.5.1 i 5.5.2 prikazani su rezultati simulacija razmatranih neispravnosti na primeru jedne bazne radne tačke za nekoliko različitih slučajeva održavanja koeficijenta viška vazduha na nivou željene vrednosti. Iako je u pitanju samo jedna bazna radna tačka motora, ipak su u dovoljnoj meri ilustrovane mogućnosti simulatora neispravnosti. Naravno, u okviru ovog istraživanja izvršene su slične simulacije u velikom broju drugih baznih radnih tačaka i ispostavilo se da pravilnosti koje su uočene u odeljcima 5.5.1 i 5.5.2 uglavnom mogu da se prenesu i na te druge radne tačke. Sporadično se u nekim baznim radnim tačkama mogu javiti neke manje specifičnosti, ali bi to trebalo da bude uzeto u obzir u dijagnostičkom modelu u procesu njegovog razvoja. Sa dijagnostičkog aspekta, bitno je da simulirane neispravnosti utiču na posmatrane karakteristične promenljive, a to je u manjoj ili većoj meri pokazano na datim primerima rezultata simulacija. To daje nadu da bi dijagnostički model, zasnovan na podacima dobijenim simulacijama neispravnosti, mogao da dâ zadovoljavajuće rezultate u smislu otkrivanja i prepoznavanja neispravnosti razmatranih tipova. U sledećem poglavlju se prikazuje, na koji način su rezultati simulatora neispravnosti iskorišćeni u razvoju tog dijagnostičkog modela.

## 6 DIJAGNOSTIČKI MODEL BAZIRAN NA REZULTATIMA SIMULACIJA

### 6.1 Uvodne napomene

Dijagnostički model, koji će biti predstavljen u ovom poglavlju, razvijan je uz sledeća ograničenja:

- motor, čiji je usisni sistem predmet dijagnostike, je u automobilu sa konvencionalnim menjačkim prenosnikom,
- broj stepeni prenosa menjačkog prenosnika nije veći od pet.

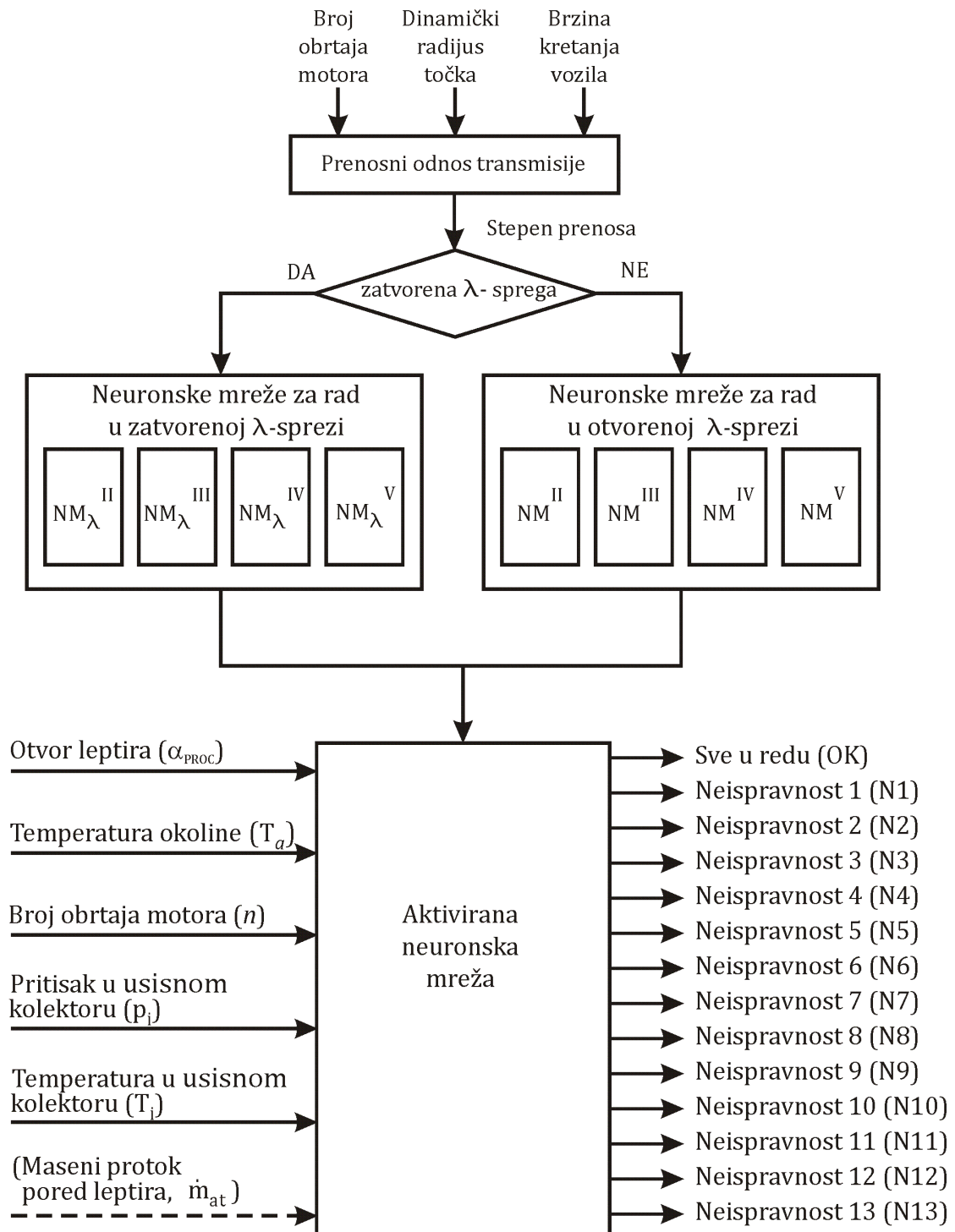
Prvo ograničenje je apsolutno, tj. razvijeni model ne može da se primeni na automobile sa kontinualnim menjačkim prenosnikom. Drugo je relativno i nije tako strogo, jer bi ovaj model mogao pod određenim uslovima da se primeni za bilo koji diskretan broj stepeni prenosa.

Ovaj model pripada grupi tzv. *modela zasnovanih na podacima* (eng. *data-driven models*). To znači da njegovu osnovu ne čine jednačine bazirane na fizičkim zakonitostima koje povezuju relevantne veličine u usisnom sistemu motora, nego veliki broj podataka o tim veličinama. Tako veliki broj podataka može se dobiti samo eksperimentalnim putem – bilo u realnim ili u virtuelnim eksperimentima. Konkretno, u ovom poglavlju biće prikazan dijagnostički model koji je razvijen na bazi podataka, dobijenih u virtuelnim eksperimentima pomoću simulatora neispravnosti, opisanog u Poglavlju 5. U Poglavlju 7 biće prikazan i dijagnostički model, razvijen na osnovu podataka dobijenih eksperimentalnim istraživanjima u laboratoriji.

### 6.2 Struktura i koncept funkcionisanja modela

Razvijeni dijagnostički model koristi tehniku neuronskih mreža (NM) za otkrivanje i prepoznavanje neispravnosti, tako da je on u stvari i sačinjen od neuronskih mreža. Da bi se čitaocu malo približio sam model, njegov uprošćeni šematski prikaz dat je na *Sl. 6.1*. Na osnovu pomenutog prikaza može se steći i uvid u to, kako bi, na konceptualnom nivou, jedan takav dijagnostički model mogao da funkcioniše u vozilu.

Kako je prikazano na *Sl. 6.1*, suštinu modela čini osam nezavisnih neuronskih mreža – četiri za rad motora u zatvorenoj sprezi sa lambda-sondom i još četiri za rad u otvorenoj sprezi sa lambda-sondom. Brojka „četiri“ ovde označava da će u svakom od pomenuta dva režima biti interesantna po četiri stepena prenosa: II, III, IV i V. Prvi stepen prenosa ovde nije razmatran, jer je udeo korišćenja tog stepena prenosa tokom eksploatacije vozila zanemariv. Neuronske mreže koje mogu da se aktiviraju samo pri radu u režimu zatvorene sprege sa lambda-sondom, označene su sa  $NM_{\lambda}$ , a neuronske mreže koje mogu da se aktiviraju samo pri radu motora u otvorenoj sprezi sa lambda-sondom, označene su samo sa NM. Rimski broj u eksponentu svake oznake neuronske mreže označava pri kojem stepenu prenosa vozila se aktivira konkretna neuronska mreža.



Sl. 6.1 Blok-šema principa rada dijagnostičkog modela

Sl. 6.1 pokazuje i kako bi ovaj model u načelu trebalo da funkcioniše. Na osnovu broja obrtaja motora  $n$  [ $\text{min}^{-1}$ ], dinamičkog radijusa točka  $r_D$  [ $m$ ] i brzine kretanja vozila  $v$  [ $km/h$ ], ECU određuje trenutni ukupni prenosni odnos transmisije  $i_T$ , po formuli  $i_T = \frac{0,12 \cdot \pi \cdot r_D \cdot n}{v}$ . Koristeći izračunati prenosni odnos transmisije, ECU prepoznaje u kom stepenu prenosa se vozilo trenutno kreće. Dalje, ECU ima informaciju da li motor trenutno radi u režimu zatvorene ili otvorene sprege sa lambda-sondom. Ove dve informacije određuju koja od osam neuronskih mreža će biti aktivirana. Kada se aktivira odgovarajuća neuronska mreža, ona prihvata sledeće ulazne podatke:

- otvor leptira u procentima ( $\alpha_{PROC}$ ),
- temperatura okoline,  $T_a$ ,
- broj obrtaja motora,  $n$ ,
- pritisak u usisnom kolektoru,  $p_i$ ,
- temperatura u usisnom kolektoru,  $T_i$  i
- maseni protok vazduha,  $\dot{m}_{at}$ , (samo ukoliko se radi o motoru sa masenim protokomerom vazduha),

pri čemu su to podaci koje ECU sračunava na osnovu signala odgovarajućih senzora.

Kako je prikazano na *Sl. 6.1*, svaka neuronska mreža ima 14 izlaza, pri čemu prvi odgovara stanju kada motor radi bez neispravnosti, a od preostalih 13, svaki odgovara po jednoj od neispravnosti, pobrojanih ranije u *Tab. 5.6*.

Dakle, za svaki ulazni uzorak ( $\alpha_{PROC}$ ,  $T_a$ ,  $n$ ,  $p_i$ ,  $T_i$ ,  $\dot{m}_{at}$ ), neuronska mreža daje izlazni vektor (OK, N1, N2, N3, N4, N5, N6, N7, N8, N9, N10, N11, N12, N13). Na osnovu vrednosti svih elemenata izlaznog vektora, određuje se trenutno dijagnostičko stanje za usisni sistem motora. Drugim rečima, aktivirana neuronska mreža prepoznaje da li motor trenutno radi sa nekom od predefinisanih neispravnosti ili je sve u redu. U nekom sledećem vremenskom trenutku novi ulazni uzorak stiže na ulaz aktivirane neuronske mreže, koja posle obrade ulaznog uzorka daje novi izlazni vektor. Na osnovu izlaznog vektora određuje se novo dijagnostičko stanje usisnog sistema, posle čega stiže sledeći uzorak na ulaz neuronske mreže i tako dalje. Paralelno sa ovim procesom, ECU izračunava trenutni prenosni odnos transmisije i proverava da li motor radi u režimu povratne sprege sa lambda-sondom, kako bi mogla da aktivira neku drugu neuronsku mrežu, ako je potrebno.

Ovaj koncept se može primeniti samo ako su ispunjeni određeni uslovi. Ukoliko je temperatura rashladne tečnosti motora ispod deklarisanе radne temperature ili ECU registruje ubrzavanje ili usporavanje motora, predloženi dijagnostički model je neaktivan. Do aktiviranja modela dolazi, tek kada se dostigne radna temperatura motora i kada se broj obrtaja stabilizuje na nekoj vrednosti, sa oscilacijama do oko  $\pm 15 \text{ min}^{-1}$ , što se može smatrati ustaljenim radom motora. To znači da bi ECU trebalo da prati temperaturu rashladne tečnosti, kao i promenu broja obrtaja motora, što je izvodljivo na današnjim automobilima.

Suštinu predloženog dijagnostičkog modela, dakle, čini osam neuronskih mreža. Podaci za ove neuronske mreže obezbeđeni su pomoću simulatora neispravnosti u usisnom sistemu, koji je takođe razvijen u okviru ovog istraživanja, a opisan je u Poglavlju 5. Kako je tamo pokazano, pomenuti simulator je univerzalan u tom smislu, što može da simulira neispravnosti i kod motora koji rade na principu merenja masenog protoka vazduha (motori sa masenim protokomerom) i kod motora koji rade na principu računanja masenog protoka vazduha (motori bez protokomera). U cilju održavanja kontinuiteta u odnosu na Poglavlje 5, razvijeni su dijagnostički modeli za oba pomenuta slučaja, svaki sa po osam neuronskih mreža. Ovi dijagnostički modeli su veoma slični. Osovna razlika je u broju ulaznih veličina, a postoji i mala razlika u izlaznim veličinama. Kod neuronskih mreža iz dijagnostičkog modela za motor sa masenim protokomerom vazduha (MAF senzor) među ulaznim veličinama nalazi se, između ostalih, i maseni protok vazduha, dok ga neuronske mreže iz dijagnostičkog modela za motor bez protokomera, nemaju. Brojevi izlaznih veličina su isti kod neuronskih mreža oba modela, ali ne postoji potpuna podudarnost između njih, jer razmatrane neispravnosti kod ta dva modela nisu baš potpuno iste (videti *Tab. 5.6*).

Zbog ovako malih razlika između dva razvijena dijagnostička modela, oni neće biti pojedinačno opisivani. U nastavku će biti prikazana realizacija modela samo za slučaj motora sa masenim protokomerom vazduha, a tamo gde je potrebno, biće date dopune koje se odnose na model za motore bez protokomera vazduha.

## 6.3 Razvoj neuronskih mreža za dijagnostiku neispravnosti u usisnom sistemu

Realizacija neuronskih mreža za otkrivanje i prepoznavanje neispravnosti u usisnom sistemu motora tekla je prema proceduri, opisanoj u odeljku 3.5 i sažeto prikazanoj na *Sl. 3.7*. U nastavku se opisuje kako je u ovom konkretnom slučaju sprovedena svaka faza te procedure. Treba napomenuti da je realizacija svih neuronskih mreža sprovedena korišćenjem programskog okruženja Matlab 2010a, u kojem su izrađeni svi programi, neophodni za ostvarenje postavljenog cilja.

### 6.3.1 Definisane ulaznih i izlaznih veličina

Pošto se ovde radi o razvoju neuronskih mreža, baziranih na podacima dobijenim u virtuelnim eksperimentima pomoću simulatora neispravnosti, samo one veličine koje su određivane u tim eksperimentima dolaze u obzir pri definisanju ulaznih i izlaznih veličina.

Najpre će biti definisane izlazne veličine. S obzirom da je u pitanju zadatak dijagnostikovanja neispravnosti, veoma je pogodno da broj izlaznih veličina bude jednak broju stanja usisnog sistema (sa stanovišta neispravnosti), tako da svaka izlazna veličina reprezentuje po jedno stanje. Skup stanja usisnog sistema za slučaj motora sa masenim protokomerom vazduha, zajedno sa nazivima odgovarajućih izlaznih veličina, dat je u *Tab. 6.1*, a za motor bez protokomera vazduha, u *Tab. 6.2* Oznake izlaznih veličina odabrane su tako da budu deskriptivne, kako bi se vrlo lako prepoznalo, koje stanje reprezentuje konkretna veličina.

*Tab. 6.1 Izlazne veličine neuronskih mreža – slučaj sa masenim protokomerom vazduha*

Stanje usisnog sistema	Izlazne veličine neuronskih mreža		
	Oznaka	Vrednost ako sistem jeste u tom stanju	Vrednost ako sistem nije u tom stanju
Nema neispravnosti (Sve OK)	„Sve OK“	1	0
MAP senzor očitava za 10 kPa nižu vrednost	„MAP -10“	1	0
MAP senzor očitava za 5 kPa nižu vrednost	„MAP -5“	1	0
MAP senzor očitava za 5 kPa višu vrednost	„MAP +5“	1	0
MAP senzor očitava za 10 kPa višu vrednost	„MAP +10“	1	0
MAF senzor očitava 10 % nižu vrednost	„MAF 90%“	1	0
MAF senzor očitava 5 % nižu vrednost	„MAF 95%“	1	0
MAF senzor očitava 5 % višu vrednost	„MAF 105%“	1	0
MAF senzor očitava 10 % višu vrednost	„MAF 110%“	1	0
Ulaz „falš“-vazduha kroz otvor prečnika 2mm	„FaV 2mm“	1	0
Ulaz „falš“-vazduha kroz otvor prečnika 4mm	„FaV 4mm“	1	0
Ulaz „falš“-vazduha kroz otvor prečnika 6mm	„FaV 6mm“	1	0
EGR protok umanjen na 80%	„EGR 80%“	1	0
EGR protok umanjen na 60%	„EGR 60%“	1	0



Tab. 6.2 Izlazne veličine neuronskih mreža – slučaj bez protokomera vazduha

Stanje usisnog sistema	Izlazne veličine neuronskih mreža		
	Oznaka	Vrednost ako sistem jeste u tom stanju	Vrednost ako sistem nije u tom stanju
Nema neispravnosti (Sve OK)	„Sve OK“	1	0
MAP senzor očitava za 10 kPa nižu vrednost	„MAP -10“	1	0
MAP senzor očitava za 5 kPa nižu vrednost	„MAP -5“	1	0
MAP senzor očitava za 5 kPa višu vrednost	„MAP +5“	1	0
MAP senzor očitava za 10 kPa višu vrednost	„MAP +10“	1	0
IAT senzor očitava za 15 stepeni nižu vrednost	„IAT -15“	1	0
IAT senzor očitava za 10 stepeni nižu vrednost	„IAT -10“	1	0
IAT senzor očitava za 10 stepeni višu vrednost	„IAT +10“	1	0
IAT senzor očitava za 15 stepeni višu vrednost	„IAT +15“	1	0
Ulaz „falš“-vazduha kroz otvor prečnika 2mm	„FaV 2mm“	1	0
Ulaz „falš“-vazduha kroz otvor prečnika 4mm	„FaV 4mm“	1	0
Ulaz „falš“-vazduha kroz otvor prečnika 6mm	„FaV 6mm“	1	0
EGR protok umanjen na 80%	„EGR 80%“	1	0
EGR protok umanjen na 60%	„EGR 60%“	1	0

U slučaju da nema neispravnosti u usisnom sistemu, tada izlazna veličina „SveOK“ ima vrednost 1, a inače ima vrednost 0. Isto pravilo važi i za sve ostale izlazne veličine, tj. ako postoji neka neispravnost u usisnom sistemu, onda izlazna veličina koja reprezentuje tu neispravnost, ima vrednost 1, a inače ima vrednost 0. Budući da je jedna od osnovnih pretpostavki ovog rada ta, da u bilo kom trenutku može postojati najviše jedna neispravnost, to znači da, kada jedna izlazna veličina ima vrednost 1, sve ostale treba da budu jednake nuli. Na primer, ako MAP senzor očitava za 10 kPa nižu vrednost, onda izlaz neuronske mreže treba da bude (0,1,0,0,0,0,0,0,0,0,0,0,0,0,0) ili ako očitava 5 kPa nižu vrednost, onda izlaz neuronske mreže treba da bude (0,0,1,0,0,0,0,0,0,0,0,0,0,0,0).

Inače, iz Tab. 6.1 i Tab. 6.2 vidi se ranije pomenuta mala razlika između dva razvijena modela, a to je da su izlazne veličine „MAF 90%“, „MAF 95%“, „MAF 105%“ i „MAF 110%“ iz prve tabele, zamenjene izlaznim veličinama „IAT -15“, „IAT -10“, „IAT +10“ i „IAT +15“ u drugoj tabeli. Ove razlike proizilaze iz činjenice da, kod motora koji nemaju maseni protokomer vazduha, dijagnostikovanje grešaka očitavanja masenog protokomera vazduha nema smisla, dok dijagnostikovanje grešaka očitavanja senzora temperature u usisnom sistemu, itekako ima smisla.

Za razliku od izlaznih veličina, prilikom čijeg odabira je sve bilo prilično jasno, u slučaju ulaznih veličina, situacija je nešto složenija. Osnovna ideja pri definisanju ulaznih veličina za neuronske mreže zasniva se na sledećim razmatranjima.

S obzirom da predloženi dijagnostički modeli treba da funkcionišu na ustaljenim režimima rada motora iz njegovog realnog radnog opsega, podaci za obuku neuronskih mreža moraju dobro „pokriti“ taj radni opseg. To znači da neuronska mreža mora biti „upoznata“ sa radom motora u većem broju baznih radnih tačaka. Odatle se zaključuje, da bi među ulaznim veličinama neuronske mreže, morale da se nađu veličine koje definišu baznu radnu tačku motora pri eksploataciji u vozilu. To su *otvor leptira*, *stepen prenosa* u kojem se vozilo kreće i *uzdužni nagib puta*. S obzirom da se po jedna neuronska mreža formira za svaki *stepen prenosa*, onda se stepen prenosa neće pojaviti među ulaznim veličinama neuronskih mreža.

Osim veličina koje definišu baznu radnu tačku motora, neuronskoj mreži koja treba da dijagnostikuje neispravnosti u radu motora, na ulazu su potrebne i veličine koje su osetljive na pojavu tih neispravnosti u što većoj meri. Drugim rečima, za ulazne veličine pogodne su one veličine, za koje se smatra da će pretrpeti dovoljno veliku promenu vrednosti, kada motor radi uz prisustvo neke od neispravnosti, pobrojanih u *Tab. 6.1* ili *Tab. 6.2* (zavisno od vrste motora). Konkretniji uvid u ponašanje pojedinih veličina pri radu motora sa nekom neispravnošću, može se steći iz odeljka 5.5, gde su prikazani i analizirani rezultati dobijeni u virtuelnim eksperimentima pomoću simulatora neispravnosti. Tamo su date promene vrednosti pojedinih veličina pri svim razmatranim neispravnostima, u jednoj baznoj radnoj tački, ali za obe razmatrane vrste motora i pri oba razmatrana lambda-režima. Iz analize tih rezultata, ali i rezultata u drugim baznim radnim tačkama, može se zaključiti da bi kao ulazne veličine neuronskih mreža trebalo da figurišu *broj obrtaja motora*, *pritisak* i *temperatura u usisnom kolektoru* i *maseni protok vazduha*. Naravno, ukoliko se radi o motoru koji nema maseni protokomer vazduha, onda promenljivu *maseni protok vazduha* treba izostaviti.

Izvođenjem virtuelnih eksperimenata za uslove različitih vrednosti temperature okolnog vazduha, ustanovljeno je da uticaj spoljašnje temperature na temperaturu u usisnom kolektoru nije zanemariv. Razlog tome je relativno širok raspon vrednosti temperature okolnog vazduha (u našim krajevima i do 50 stepeni), koji se meša u usisnom kolektoru sa gasovima iz EGR sistema. Temperatura izduvnih gasova je znatno viša od temperature okolnog vazduha, ali je dotok spoljašnjeg vazduha u usisni kolektor znatno veći od dotoka EGR gasova. Otuda će hladan zimski vazduh koji ulazi u usisni kolektor uzrokovati приметно nižu temperaturu u usisnom kolektoru u odnosu na topao letnji vazduh. Kada se ovoj diskusiji dodaju činjenice da su među razmatranim neispravnostima i pogrešno očitavanje senzora temperature u usisnom kolektoru, kao i umanjenje EGR protoka, sam se nameće zaključak da bi jedna od ulaznih veličina neuronskih mreža dijagnostičkog modela trebalo da bude i *temperatura okolnog vazduha*. U prilog ovom zaključku ide i činjenica da spoljašnja temperatura, preko gustine vazduha utiče u određenoj meri i na punjenje motora, tako da se temperatura okolnog vazduha može smatrati dopunskom promenljivom koja utiče na definisanje radne tačke motora.

Sumirajući zaključke razmatranja iz prethodnih nekoliko pasusa, mogla bi se napraviti lista ulaznih veličina za neuronske mreže dijagnostičkih modela. Međutim, jedan tehnički problem koji se pojavio prilikom opsežnih preliminarnih istraživanja, uticao je da ta lista bude skraćena za jednog člana. Naime, ustanovljeno je da se variranjem *otvora leptira*, *uzdužnog nagiba puta* i *temperature okolnog vazduha* dobijaju veoma veliki skupovi podataka za obuku neuronskih mreža. Da bi se ovi skupovi sveli na razumnu meru, odlučeno je da se jedna od tri pobrojane veličine odstrani iz skupa ulaznih veličina. *Otvor leptira* je najvažniji parametar i nikako ne može biti izostavljen iz skupa ulaznih veličina neuronskih mreža. *Temperatura okolnog vazduha* je veoma važna za otkrivanje neispravnosti senzora temperature u usisnom kolektoru i neispravnosti u EGR sistemu, pa je neophodno i prisustvo ove veličine na ulazu u neuronske mreže. *Uzdužni nagib puta* je parametar, čijim variranjem se proširuje radni opseg motora, pa je najbezbolnije da se uvede uprošćenje o kretanju vozila po horizontalnom putu. Drugim rečima, uzdužni nagib puta će pri svim virtuelnim eksperimentima biti jednak nuli i neće se pojaviti kao ulazna veličina neuronske mreže.

Na kraju, posle pomenute korekcije, skup ulaznih promenljivih postaje konačan i prikazan je u *Tab. 6.3*. Poslednja ulazna veličina u tabeli, *maseni protok vazduha*, stavljena je u zagradu, jer se pojavljuje kao ulazna veličina samo kod neuronskih mreža dijagnostičkog modela za slučaj motora sa masenim protokomerom vazduha.

Tab. 6.3 Ulazne promenljive neuronskih mreža dijagnostičkog modela

Ulazna promenljiva neuronske mreže	
Naziv	Oznaka
Otvor leptira u procentima	$\alpha_{PROC}$
Temperatura okolnog vazduha	$T_a$
Broj obrtaja motora	$n$
Pritisak u usisnom kolektoru	$p_i$
Temperatura u usisnom kolektoru	$T_i$
(Maseni protok vazduha)	$(\dot{m}_{at})$

### 6.3.2 Prikupljanje i priprema podataka za obuku

Kao što je već više puta napomenuto, kao izvor podataka za obuku korišćen je simulator neispravnosti, razvijen takođe u okviru ovog istraživanja i detaljno opisan u Poglavlju 5. Ovde treba podsetiti da se pri jednom izvršavanju simulatora neispravnosti, ostvare simulacije svih 14 stanja usisnog sistema u jednoj baznoj radnoj tački, definisanoj otvorom leptira, stepenom prenosa u kojem se vozilo kreće i temperaturom okolnog vazduha.

Broj baznih radnih tačaka, u kojima tokom eksploatacije radi automobilski motor, je ogroman, pri čemu se one kreću u širokom radnom opsegu. Zadatak je da se izabere razuman broj baznih radnih tačaka motora, da se ne bi dobio preveliki skup podataka za obuku, koji bi prevazišao mogućnosti raspoloživih računarskih resursa. Sa druge strane, izabrane bazne radne tačke treba što bolje da reprezentuju radni opseg automobilskog motora. Da bi se u što većoj meri zadovoljila ova dva suprotstavljena zahteva, odlučeno je da se posmatrani radni opseg suzi, ali tako da u što većoj meri bude pokriven onaj deo celokupnog radnog opsega motora, u kojem automobilski motor najčešće radi tokom eksploatacije. U tom smislu, pretpostavljeni su orijentacioni opsezi brzina kretanja automobila u pojedinim stepenima prenosa:

- II stepen prenosa, od 25 do 40 km/h,
- III stepen prenosa, od 40 do 60 km/h,
- IV stepen prenosa, od 60 do 90 km/h i
- V stepen prenosa, od 80 do 120 km/h.

Zatim su određeni odgovarajući opsezi broja obrtaja motora, na osnovu dinamičkog radijusa točka i podataka o prenosnim odnosima transmisije za pojedine stepene prenosa iz Tab. 5.1. Posle toga, izvođeni su brojni preliminarni virtuelni eksperimenti pomoću simulatora neispravnosti, kako bi se ustanovili opsezi otvora leptira po pojedinim stepenima prenosa, koji će približno da „pokriju“ gore navedene opsege brzina automobila. Na osnovu svega toga, usvojen je plan izvođenja virtuelnih eksperimenata, koji je predstavljen u Tab. 6.4. Tabela pokazuje da je za prikupljanje podataka za obuku svake neuronske mreže izvedeno po 20 virtuelnih eksperimenata, što znači da su simulacije izvedene u 20 različitih baznih tačaka za svaki stepen prenosa. Tih 20 baznih radnih tačaka ostvareno je izvođenjem simulacija za pet različitih otvora leptira na četiri različite temperature okoline. Opseg otvora leptira usvojen je kako je objašnjeno malo ranije u ovom pasusu, a opseg temperatura okoline od 0 do 30°C usvojen je tako da odražava realne temperature u našem podneblju. Do broja od 4 različite vrednosti temperature okoline i 5 različitih vrednosti otvora leptira došlo se iskustveno tokom preliminarnih istraživanja.



	54			12		134		12
	55			13		135		13
	56			9		136		9
	57			10		137		10
	58		30	11		138	30	11
	59			12		139		12
	60			13		140		13
NM <sub>λ</sub> <sup>V</sup>	61	V	0	11	NM <sup>V</sup>	141	V	11
	62			13		142		13
	63			15		143		15
	64			17		144		17
	65			20		145		20
	66		10	11		146		11
	67			13		147		13
	68			15		148		15
	69			17		149		17
	70			20		150		20
	71		20	11		151		11
	72			13		152		13
	73			15		153		15
	74			17		154		17
	75			20		155		20
	76		30	11		156		11
	77			13		157		13
	78			15		158		15
	79			17		159		17
	80			20		160		20

Dakle, za obuku neuronskih mreža dijagnostičkog modela usisnog sistema sa masenim protokomerom bilo je potrebno izvesti 160 virtuelnih eksperimenata (20 baznih radnih tačaka x 4 stepena prenosa x 2  $\lambda$ -režima). Za dijagnostički model usisnog sistema bez protokomera, virtuelni eksperimenti su izvođeni prema istom planu (Tab. 6.4). To znači da ukupan broj virtuelnih eksperimenata za oba dijagnostička modela iznosi 320.

Da bi slika o prikupljanju podataka za obuku bila potpunija, ovde još treba podsetiti da se kao glavni rezultat jednog virtuelnog eksperimenta pomoću programa **NSimulatorMenadzer** dobija jedan Excel-fajl sa tabelom ili matricom, čiji oblik je ilustrovan ranije na strani 94, Tab. 5.7. U kolonama matrice su ulazne veličine neuronske mreže, a u vrstama su vrednosti tih veličina tokom izvođenja simulacija 14 posmatranih stanja usisnog sistema. Prilikom simulacije svakog stanja zapamćeno je po 13 simulacionih uzoraka, a zašto baš taj broj, detaljno je objašnjeno ranije, u odeljku 5.4.3. Dakle, ukupan broj vrsta pomenute matrice iznosi 182 (14 stanja x 13 simulacionih uzoraka), a to je, u stvari, broj uzoraka za obuku koji se dobijaju pri jednom virtuelnom eksperimentu. Pošto je za obuku jedne neuronske mreže potrebno sakupiti podatke iz 20 virtuelnih eksperimenata, to znači da će skup podataka za obuku sadržati ukupno 3640 uzoraka (182x20). Kad se još uzme u obzir broj ulaznih veličina neuronske mreže, može se zaključiti da će podaci za obuku stati u matricu formata 3640x6 za slučaj usisnog sistema sa masenim protokomerom, odnosno 3640x5 za usisni sistem bez protokomera.

Međutim, pomenuta matrica ne sadrži sve podatke koji su potrebni za obuku neuronske mreže, već sadrži samo ulaznu komponentu skupa za obuku (videti Sl. 3.4 i prateće objašnjenje). Zbog toga će, u daljem delu teksta, ova matrica nositi naziv *ulazna matrica za obuku*. Neophodna je još i izlazna komponenta skupa za obuku odnosno *matrica željenih izlaza* neuronske mreže. Ova matrica mora imati isti broj vrsta kao i ulazna matrica za obuku, a broj njenih kolona mora biti jednak broju izlaznih veličina neuronske mreže. Dakle, format matrice željenih izlaza neuronske mreže u konkretnom slučaju je 3640x14, a ona je specifična po tome, što će u svim svojim vrstama imati 13 nula i jednu jedinicu.

Pozicija te jedinice u nekoj vrsti matrice željenih izlaza, menjaće se u zavisnosti od stanja usisnog sistema koje reprezentuje odgovarajuća vrsta u ulaznoj matrici za obuku. Da bi ovo bilo jasnije, na Sl. 6.2 daje se ilustracija matrice željenih izlaza.

14 kolona

		SveOK	MAP-10	MAP-5	.	.	.	EGR60%
Virtuelni eksperiment br. 1 (14 x 13 vrsta)	13 vrsta	1	0	0	.	.	.	0
		1	0	0	.	.	.	0
		...	...	...	...	...	...	...
	13 vrsta	1	0	0	.	.	.	0
		0	1	0	.	.	.	0
		0	1	0	.	.	.	0
	13 vrsta	0	1	0	.	.	.	0
		0	0	1	.	.	.	0
		0	0	1	.	.	.	0
	13 vrsta	0	0	1	.	.	.	0
		0	0	0	.	.	.	1
		0	0	0	.	.	.	1
	Virtuelni eksperiment br. 20 (14 x 13 vrsta)	13 vrsta	1	0	0	.	.	.
1			0	0	.	.	.	0
...			...	...	...	...	...	...
13 vrsta		1	0	0	.	.	.	0
		0	1	0	.	.	.	0
		0	1	0	.	.	.	0
13 vrsta		0	1	0	.	.	.	0
		0	0	1	.	.	.	0
		0	0	1	.	.	.	0
13 vrsta		0	0	1	.	.	.	0
		0	0	1	.	.	.	0
		0	0	0	.	.	.	1
13 vrsta		0	0	0	.	.	.	1
	0	0	0	.	.	.	1	
	0	0	0	.	.	.	1	
3640 x 14								

Sl. 6.2 Ilustracija matrice željenih izlaza neuronske mreže

Da bi se kreirala jedna ovakva matrica, napisana je jedna program-funkcija, pod nazivom *MatricaZeljenihIzlaza*, čiji je izvorni kod dat u prilogu P3.1. Ova funkcija je univerzalna, u smislu da joj samo treba proslediti dva parametra – proizvoljan broj stanja usisnog sistema

i proizvoljan broj snimljenih uzoraka po simulaciji jednog stanja - i ona će generisati odgovarajuću matricu željenih izlaza neuronske mreže.

Funkcija *MatricaZeljenihIzlaza* poziva se u okviru jednog drugog, većeg programa, koji se zove *generisiStrukturuZaObuku*. Program *generisiStrukturuZaObuku* omogućava da se u jednom Matlab objektu, koji se zove Matlab Struktura, objedine svi podaci potrebni za obuku jedne neuronske mreže. To su, pre svega, ulazna matrica za obuku i matrica željenih izlaza, koje su opisane u ovom odeljku. Ulazna matrica za obuku generiše se iz proizvoljnog broja excel fajlova, dobijenih u virtuelnim eksperimentima, a matrica željenih izlaza, kako je malo pre rečeno, generiše se pozivanjem programa *MatricaZeljenihIzlaza*. Izvorni kod programa *generisiStrukturuZaObuku* može se videti u prilogu P3.2, a da bi se sagledala suština programa, ovde se daje njegov pseudokod:

### **Program generisiStrukturuZaObuku**

#### **Počni**

**Učitaj broj stanja** usisnog sistema

**Učitaj** željene Excel fajlove u niz Excel fajlova

**Inicijalizuj** praznu ulaznu matricu za obuku

**Inicijalizuj** naziv Matlab fajla

**Postavi brojač** Excel fajlova na 1

**Dok brojač** excel fajlova ne premaši dužinu niza Excel fajlova, **radi**

**Preuzmi** podatke iz *tekućeg Excel fajla* u *tekuću matricu*

**Nadoveži** po vertikali *tekuću matricu* na *ulaznu matricu za obuku*

**Ekstrahuj** relevantne podatke iz imena *tekućeg Excel fajla*

**Pridodaj** relevantne podatke *nazivu Matlab fajla*

**Povećaj brojač** Excel fajlova za 1

**Odredi broj uzoraka** po stanju kao

$$\text{broj uzoraka po stanju} = \frac{\text{Broj vrsta ulazne matrice za obuku}}{\text{broj stanja} \cdot \text{broj Excel fajlova}}$$

**Generiši** matricu željenih izlaza pozivanjem funkcije

*MatricaZeljenihIzlaza(broj stanja, broj uzoraka po stanju)*

**Objedini** broj Excel fajlova, ulaznu matricu za obuku i matricu željenih izlaza u jednu strukturu za obuku

**Generiši** konačan naziv Matlab fajla

**Snimi** strukturu za obuku u Matlab fajl

#### **Završi**

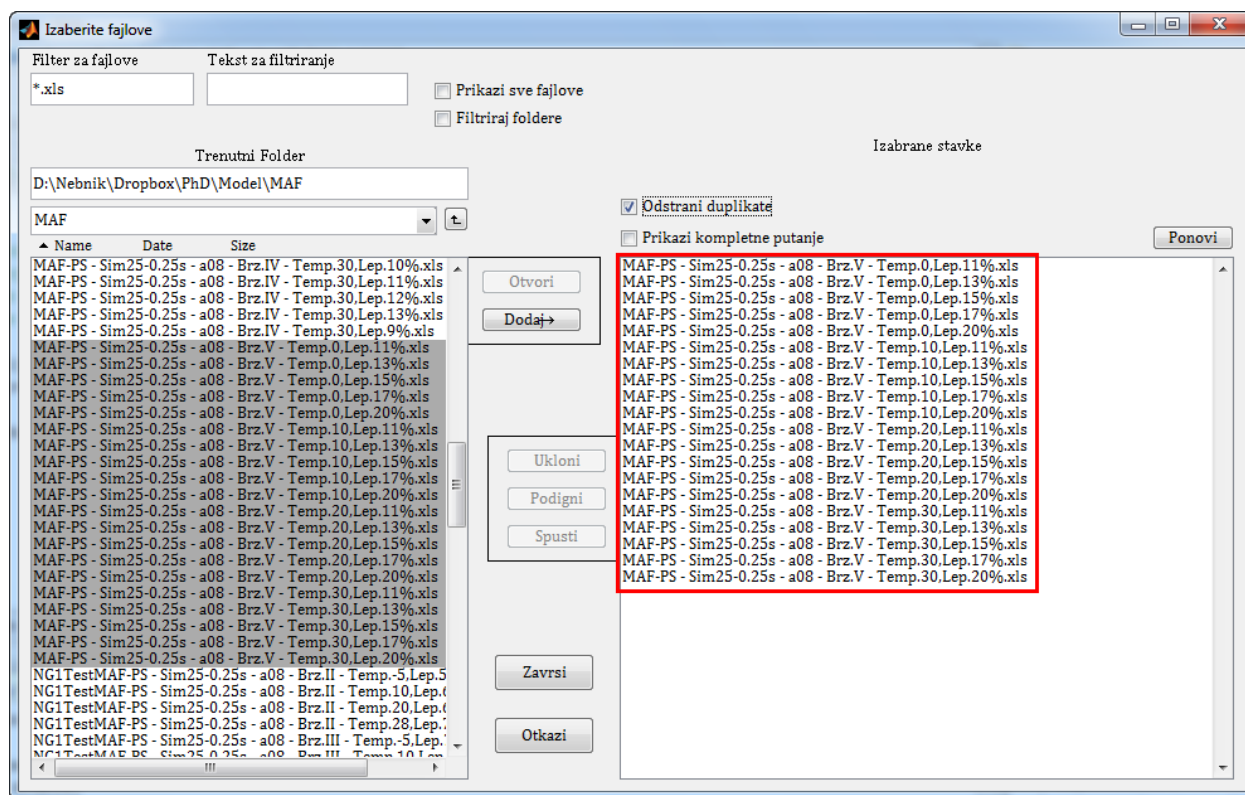
Korišćenjem programa *generisiStrukturuZaObuku*, na veoma lak način se može dobiti skup podataka za obuku neuronske mreže, bez obzira na:

- broj simuliranih stanja usisnog sistema,
- broj simulacionih uzoraka po stanju i
- broj radnih tačaka u kojima su izvođeni virtuelni eksperimenti.

Ovo svojstvo programa *generisiStrukturuZaObuku* pokazalo se posebno korisnim u fazi preliminarnih istraživanja, kada još nisu bili utvrđeni ni konačan broj stanja usisnog sistema, ni vreme trajanja simulacija, ni interval uzorkovanja, ni broj virtuelnih eksperimenata. Postojala je potreba za variranjem svih ovih, a i nekih drugih parametara, što je je iziskivalo kreiranje jako velikog broja prilično glomaznih skupova za obuku. Zbog

toga je manuelna realizacija ove faze razvoja neuronske mreže bila praktično nemoguća, pa se izrada programa kao što je *generisiStrukturuZaObuku* nametnula kao obavezna.

Sada sledi kratka ilustracija jednostavnosti funkcionisanja programa *generisiStrukturuZaObuku* sa stanovišta korisnika. Pokretanjem ovog programa, otvara se prozor za izbor Excel-fajlova, prikazan na slici Sl. 6.3.



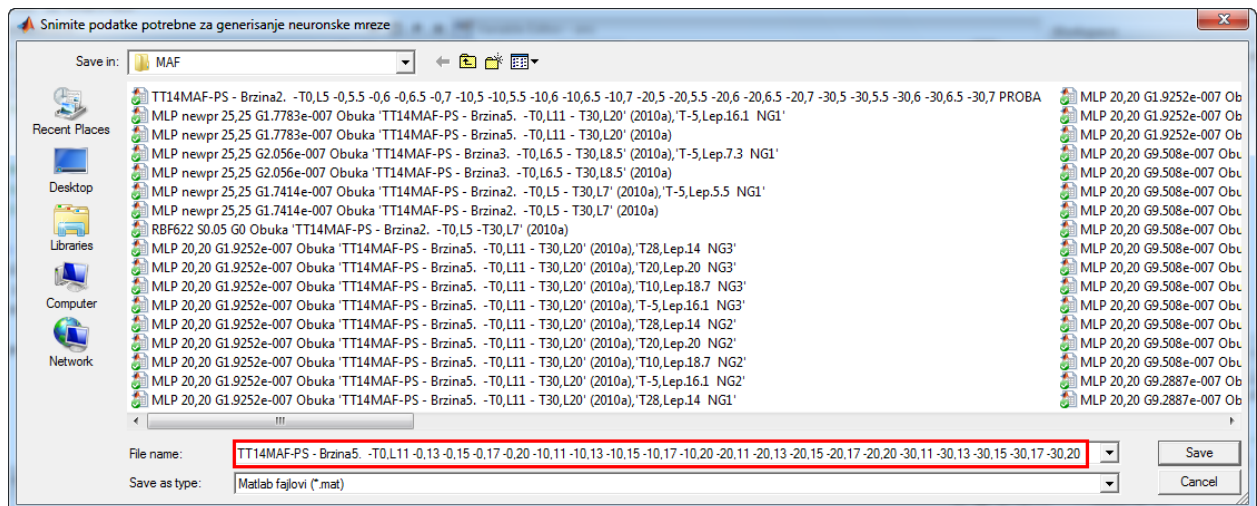
Sl. 6.3 Prozor za izbor Excel-fajlova pri generisanju skupa za obuku neuronske mreže

U tom prozoru odabere se proizvoljan broj Excel-fajlova, čiji se podaci žele koristiti u obuci buduće neuronske mreže. U primeru na slici, odabrano je 20 fajlova i oni su uokvireni crvenim pravougaonikom. Posle klika na dugme „Završi“, prozor za izbor fajlova se zatvara. Nakon toga se, prema ranije prikazanom pseudokodu, u trajanju od nekoliko sekundi odvija formiranje Matlab strukture, koja sadrži odgovarajući skup za obuku nastao od podataka iz izabranih fajlova. Na kraju, otvara se prozor za snimanje generisane Matlab strukture u fajl, čiji naziv je program automatski generisao i ponudio u odgovarajućem polju (uokvireno crvenim pravougaonikom na Sl. 6.4). Ponudeni naziv je generisan na osnovu relevantnih podataka iz svakog odabranog Excel-fajla i to tako, da se na osnovu tog naziva odmah imaju sledeće informacije:

- koliko stanja usisnog sistema je posmatrano,
- da li je u pitanju motor sa masenim protokomerom ili bez protokomera,
- da li su prikupljeni podaci dobijeni pri simulaciji rada motora u zatvorenoj ili otvorenoj sprezi sa lambda-sondom,
- na koji stepen prenosa se odnose prikupljeni podaci,
- pri kojim kombinacijama temperature okolnog vazduha i otvora leptira su izvođeni virtuelni eksperimenti.

Priča o generisanju naziva fajlova možda nije direktno vezana za temu istraživanja, ali u velikoj meri olakšava upravljanje ogromnim količinama podataka, na kojima se zasniva predloženi dijagnostički model usisnog sistema. Zbog toga je autor smatrao da na ovom mestu treba da se ukaže i na taj aspekt razvoja dijagnostičkog modela.





Sl. 6.4 Primer automatski generisanog naziva fajla za obuku neuronske mreže

Podaci iz Matlab strukture, dobijene na gore opisani način, trebalo bi još dodatno da se pripreme za obuku, kako je opisano u odeljku 3.5. To znači da bi trebalo da se izvrši predobrada podataka koja obuhvata njihovu normalizaciju i podelu na podskupove za obuku, validaciju i testiranje. Međutim, ovde to nije neophodno eksplicitno raditi, jer se procedure za predobradu podataka automatski pokreću u okviru Matlab-ovog alata za neuronske mreže, o čemu će uskoro biti još reči.

### 6.3.3 Strukturisanje neuronske mreže

Pošto su se neuronske mreže bez povratnih sprega pokazale izuzetnim pri rešavanju zadatka prepoznavanja uzoraka, odlučeno je da se i u ovom istraživanju, gde se prepoznavanje uzoraka primenjuje u dijagnostičke svrhe, ne odstupi od tog pozitivnog iskustva. Preliminarna istraživanja su izvođena sa dve najčešće primenjivane ovakve mreže: višeslojni perceptron (BP neuronska mreža) i neuronska mreža sa radijalnom baznom funkcijom (RBF neuronska mreža). BP neuronska mreža je pokazala znatno bolje rezultate, što je uticalo da se RBF neuronska mreža isključi iz daljeg istraživanja i razmatranja.

Konačna topologija neuronske mreže određena je metodom probe i greške u većem broju iteracija. Treba istaći da su ove iteracije izvođene paralelno sa fazama obuke i testiranja neuronske mreže, jer se tek posle testiranja može odrediti da li topologiju treba još korigovati ili je ona dobila svoju konačnu formu. Ova konstatacija ne važi samo za topologiju neuronske mreže. Principijelno, sva podešavanja u cilju traženja optimalnih performansi neuronske mreže izvode se kroz sprovođenje obuke i testiranja, jer se tek testiranjem može utvrditi da li je neuronsku mrežu potrebno još korigovati i u kom smeru treba da ide ta korekcija. Međutim, u ovom odeljku će biti reči samo o korekcijama koje se tiču strukturisanja neuronske mreže, a o drugim korekcijama će se govoriti u sledećem odeljku. Dakle, što se tiče topologije, krenulo se od neuronskih mreža sa jednim skrivenim slojem neurona i manjim brojem neurona u tom sloju, ali su dobijani loši rezultati, jer mreža sa malim brojem parametara nije mogla da nauči da dijagnostikuje stanja usisnog sistema na osnovu nepoznatih uzoraka. Broj neurona u skrivenom sloju je povećavan, rezultati su bili sve bolji, ali se onda došlo do tačke, kada dalje povećanje broja skrivenih neurona u jednom sloju više nije poboljšavalo sposobnost mreže da generalizuje. U tom momentu rezultati nisu bili loši, ali su istraživanja nastavljena sa dva skrivena sloja neurona, da bi se utvrdilo da li na taj način mogu još da se poboljšaju performanse neuronske mreže. Ovo se pokazalo kao korak u dobrom smeru, jer su dobijene neuronske mreže koje veoma dobro dijagnostikuju stanja usisnog sistema. Konačni broj neurona u

skrivenim slojevima je po 25 u svakom, tako da je 6-25-25-14 konačna topologija za svih 8 neuronskih mreža za slučaj sa masenim protokomerom vazduha, a 5-25-25-14 za svih 8 neuronskih mreža za slučaj bez protokomera. Inače, neuronske mreže u okviru jednog dijagnostičkog modela ne moraju da imaju iste topologije, ali s obzirom da rešavaju isti zadatak i obrađuju slične skupove podataka, nije iznenađeno što se odabrana topologija dobro pokazala u svih 16 slučajeva (8+8).

Sa izborom aktivacionih funkcija nije bilo većih problema. Za oba skrivena sloja i za izlazni sloj odabrane su bipolarne sigmoidne aktivacione funkcije, ali je bilo nekih dilema oko aktivacione funkcije izlaznog sloja. Naime, pošto se očekuje da vrednosti izlaza neuronskih mreža budu ograničene u intervalu od 0 do 1, kao kandidat za aktivacionu funkciju izlaznog sloja nametala se binarna sigmoidna funkcija, koja ima ograničen skup vrednosti takođe u intervalu  $[0,1]$ . Međutim, posle nekoliko pokušaja sa tom aktivacionom funkcijom, utvrđeno je da se dobijaju daleko lošiji rezultati, nego sa bipolarnom sigmoidnom funkcijom. Ubrzo je otkriven i razlog ovakvog ishoda, a to je da podrazumevana funkcija za normalizaciju vrednosti ulaza i izlaza BP neuronskih mreža skalira vrednosti u intervalu  $[-1,1]$ , što nije kompatibilno sa binarnom sigmoidnom funkcijom. Zato je odlučeno da se i u izlaznom sloju neurona koristi bipolarna sigmoidna funkcija.

Kvalitet neuronskih mreža je procenjivan na osnovu uspešnosti dijagnostikovanja stanja usisnog sistema. Pri tome je za izlazne veličine neuronske mreže usvojeno da vrednost 0.5 bude prag detektovanja nekog stanja i na osnovu te vrednosti će se formirati dijagnostička tvrdnja. Ova vrednost je odabrana kao sredina između krajnjih vrednosti (0 i 1) odziva izlaznih neurona. Ako bi prag detektovanja stanja bio bliže jedinici, onda se povećava mogućnost da značajan broj stanja ostane nedetektovan, a ako bi prag bio bliže nuli, to bi za posledicu imalo veći broj lažnih alarma. Otuda se vrednost 0.5 pokazala optimalnom. To znači da, ako neka izlazna veličina ima vrednost veću od 0.5, onda je usisni sistem u stanju koje reprezentuje ta promenljiva. Slično tome, ako je vrednost neke izlazne veličine ispod 0.5, onda usisni sistem nije u stanju koje ta promenljiva reprezentuje.

Praktična realizacija strukture neuronske mreže, opisane u ovom odeljku, izvedena je pomoću namenski napisanog računarskog programa *kreirajNetStrukturu*. U okviru istog programa, realizuju se obuka neuronske mreže i jedno njeno testiranje. Za sva ostala testiranja koristi se drugi program, pod nazivom *testirajNetStrukturu*. I jedan i drugi program biće predstavljeni malo kasnije, u posebnom odeljku.

### 6.3.4 Obuka neuronske mreže

Pre sprovođenja same obuke neuronske mreže, trebalo je izvršiti izbor nekih parametara. Najpre je izabran algoritam obuke.

S obzirom da je za potrebe dijagnostičkog modela izabrana BP neuronska mreža, trebalo je izabrati jednu verziju BP algoritma koji će biti korišćen za njenu obuku. U užem izboru su bili Levenberg-Marquardt-ov algoritam, elastični BP algoritam<sup>27</sup> i skalirani konjugovano gradijentni BP algoritam<sup>28</sup>. Posle isprobavanja sva tri algoritma na većem broju slučajeva, izbor je pao na *skalirani konjugovano gradijentni BP algoritam*, jer je ovaj algoritam dao bolje rezultate, nego elastični BP algoritam. Levenberg-Marquardt-ov algoritam je u par navrata izazvao probleme sa memorijskim resursima Matlab-a, pa zbog toga nije bio konkurentan.

Zatim je izabrana funkcija greške. Odlučeno je da to bude srednja kvadratna greška, jer je ona podrazumevana (eng. *default*) funkcija greške za BP neuronske mreže u Matlabovom

<sup>27</sup> eng. *resilient backpropagation algorithm*

<sup>28</sup> eng. *scaled conjugate gradient backpropagation algorithm*

alatu za neuronske mreže. Ciljna vrednost kojoj će tokom obuke da konvergira izabrana funkcija greške postavljena je na nulu. Naravno, ova ciljna vrednost pri obuci nije mogla biti dostignuta, ali je na taj način omogućeno da se proces obuke odvija, dok se ne dostigne najmanja moguća vrednost funkcije greške, pri zadatim uslovima. S obzirom na jako veliki skup raznovrsnih podataka za obuku, nije postojala opasnost od pretreniranosti<sup>29</sup> neuronskih mreža, koja inače postoji ako se pri obuci postigne veoma mala vrednost funkcije greške.

Matlabova podrazumevana vrednost koeficijenta brzine učenja, 0.01, za izabranu verziju BP algoritma nije menjana, jer proces obuke nije bio spor, a ni dobijeni rezultati nisu ukazivali da bi trebalo varirati ovaj parametar.

Takođe, podrazumevani uzajamni odnos veličina podskupova podataka za obuku, validaciju i testiranje, 60:20:20, respektivno, ostao je nepromenjen tokom razvoja neuronskih mreža dijagnostičkog modela.

Potrebno je bilo definisati vrednosti još nekoliko parametara kojima se sprečava beskonačno trajanje procesa obuke. Maksimalni broj ciklusa učenja definisan je tako da ne bude dostignut u toku obuke. Drugim rečima, cilj je bio da do prekida obuke ne dođe zbog dostizanja maksimalnog broja ciklusa učenja, nego samo zbog nemogućnosti daljeg poboljšanja performansi neuronske mreže, što bi značilo da je dostignuta minimalno moguća vrednost funkcije greške. Iz tog razloga, za maksimalni broj ciklusa učenja usvojena je vrednost 1000, što se pokazalo dovoljnim za ispunjenje pomenutog uslova. Minimalna vrednost gradijenta greške, kao i dozvoljeni broj validacionih provera, zadržani su na Matlab-ovim podrazumevanim vrednostima,  $10^{-5}$  i 6, respektivno. Ove vrednosti je moguće izmeniti, ali tokom razvoja neuronskih mreža u ovom radu, nije bilo potrebe za tim.

Posle definisanja svih potrebnih parametara, praktična obuka neuronske mreže vrši se pozivanjem odgovarajuće Matlab-ove funkcije u okviru ranije pomenutog programa *kreirajNetStrukturu*.

### 6.3.5 Testiranje neuronske mreže

Kao što je bio slučaj sa podacima za obuku neuronskih mreža, tako su i podaci za njihovo testiranje dobijeni izvođenjem virtuelnih eksperimenata pomoću programa *NSimulatorMenadzer*. Zamišljeno je da se testiranje neuronskih mreža izvrši u više nivoa, koji se međusobno razlikuju po težini sa aspekta problema dijagnostikovanja stanja usisnog sistema. Drugim rečima, podaci za testiranje na prvom nivou su takvi, da je na osnovu njih neuronskoj mreži lako da dijagnostikuje sva stanja usisnog sistema. Sa svakim sledećim nivoom, neuronskoj mreži je sve teže da dijagnostikuje sva stanja usisnog sistema na osnovu dostupnih podataka za testiranje. Da bi ovo bilo jasnije, slede opisi nivoa, na kojima su sprovedena testiranja.

#### Nivo I

Za testiranje na ovom nivou koriste se podaci, dobijeni izvođenjem novih virtuelnih eksperimenata u poznatim baznim radnim tačkama. Kad se kaže „poznatim baznim radnim tačkama“, misli se na to, da su te radne tačke poznate neuronskoj mreži. Drugim rečima, podaci snimljeni u tim radnim tačkama korišćeni su u obuci neuronske mreže. Podsećanja radi, bazna radna tačka je definisana ranije, u odeljku 5.4.2, na strani 84, sa sledeće tri veličine: otvor leptira, stepen prenosa i temperatura okoline. Da ne bude zabune, podaci dobijeni u istim baznim radnim tačkama, ali u dva različita virtuelna eksperimenta međusobno se razlikuju. Razlikuju se zato što se prilikom izvođenja svakog virtuelnog eksperimenta generišu pulsacije broja obrtaja, pritiska u usisnom kolektoru i masenog

<sup>29</sup> eng. *overtraining*

protoka vazduha, po principu slučajnih brojeva, kako je objašnjeno u odeljku 5.4.4 (Korak VI, strana 91). Otuda podaci, dobijeni u poznatoj radnoj tački, ali u novom eksperimentu, predstavljaju nove podatke za neuronsku mrežu i mogu se koristiti za njeno testiranje. Plan izvođenja virtuelnih eksperimenata za prikupljanje podataka za testiranje na ovom nivou, dat je u *Tab. 6.5*.

*Tab. 6.5 Plan izvođenja virtuelnih eksperimenata - testiranje neuronskih mreža (Nivo I)*

Rad motora u zatvorenoj sprezi sa $\lambda$ -sondom					Rad motora u otvorenoj sprezi sa $\lambda$ -sondom				
Neuronska mreža	Red. broj. eksperim.	Stepen prenosa	Temp. okoline [°C]	Otvor leptira [%]	Neuronska mreža	Red. broj. eksperim.	Stepen prenosa	Temp. okoline [°C]	Otvor leptira [%]
NM $_{\lambda}^{\text{II}}$	1	II	0	5	NM $^{\text{II}}$	9	II	0	5
	2		10	6		10		10	6
NM $_{\lambda}^{\text{III}}$	3	III	20	7	NM $^{\text{III}}$	11	III	20	7
	4		30	8		12		30	8
NM $_{\lambda}^{\text{IV}}$	5	IV	0	11	NM $^{\text{IV}}$	13	IV	0	11
	6		10	13		14		10	13
NM $_{\lambda}^{\text{V}}$	7	V	20	13	NM $^{\text{V}}$	15	V	20	13
	8		30	17		16		30	17

Prema planu iz tabele, izvršena su po dva virtuelna eksperimenta za svaku neuronsku mrežu, što čini ukupno 16 virtuelnih eksperimenata za jedan dijagnostički model. Isto toliko je urađeno i za drugi dijagnostički model, s obzirom da je ovaj plan primenjen na oba dijagnostička modela usisnog sistema (i sa masenim protokomerom i bez protokomera).

## Nivo II

Testiranje neuronske mreže na ovom nivou vrši se pomoću podataka dobijenih virtuelnim eksperimentima u nepoznatim baznim radnim tačkama. Nepoznate bazne radne tačke na ovom nivou razlikuju se od poznatih po tome, što je neuronskoj mreži nepoznat bar jedan od ova dva elementa koji definišu baznu radnu tačku: otvor leptira i/ili temperatura okoline. Plan izvođenja virtuelnih eksperimenata na ovom nivou dat je u *Tab. 6.6* i napravljen je tako da budu zastupljena sva tri moguća slučaja:

- nepoznat otvor leptira
- nepoznata temperatura okoline i
- nepoznati i otvor leptira i temperatura okoline.

*Tab. 6.6 Plan izvođenja virtuelnih eksperimenata - testiranje neuronskih mreža (Nivo II)*

Rad motora u zatvorenoj sprezi sa $\lambda$ -sondom					Rad motora u otvorenoj sprezi sa $\lambda$ -sondom				
Neuronska mreža	Red. broj. eksperim.	Stepen prenosa	Temp. okoline [°C]	Otvor leptira [%]	Neuronska mreža	Red. broj. eksperim.	Stepen prenosa	Temp. okoline [°C]	Otvor leptira [%]
NM $_{\lambda}^{\text{II}}$	1	II	0	4.6	NM $^{\text{II}}$	25	II	0	4.6
	2		-5	5.5		26		-5	5.5
	3		10	6.8		27		10	6.8
	4		36	7		28		36	7
	5		28	7.2		29		28	7.2
	6		16	6.5		30		16	6.5
NM $_{\lambda}^{\text{III}}$	7	III	0	6.7	NM $^{\text{III}}$	31	III	0	6.7
	8		-5	7.3		32		-5	7.3
	9		10	6.2		33		10	6.2
	10		36	7.5		34		36	7.5
	11		28	8.3		35		28	8.3
	12		16	8		36		16	8

Tab 6.6. (Nastavak)

NM <sub>λ</sub> <sup>IV</sup>	13	IV	0	8.8	NM <sup>IV</sup>	37	IV	0	8.8
	14		-5	10		38		-5	10
	15		10	11.5		39		10	11.5
	16		36	11		40		36	11
	17		28	10.6		41		28	10.6
	18		16	13		42		16	13
NM <sub>λ</sub> <sup>V</sup>	19	V	0	11.6	NM <sup>V</sup>	43	V	0	11.6
	20		-5	12		44		-5	12
	21		10	15.7		45		10	15.7
	22		36	13		46		36	13
	23		28	16.3		47		28	16.3
	24		16	20		48		16	20

Prema planu iz tabele, na ovom nivou je izvršeno po šest virtuelnih eksperimenata za svaku neuronsku mrežu, što čini ukupno 48 virtuelnih eksperimenata za jedan dijagnostički model. Isto toliko je urađeno i za drugi dijagnostički model, s obzirom da je ovaj plan primenjen na oba dijagnostička modela usisnog sistema (i sa masenim protokomerom i bez protokomera).

Na nivoima I i II testira se sposobnost neuronskih mreža da prepoznaju stanja usisnog sistema koja su poznata neuronskoj mreži. Kod nivoa III i IV, proverava se koliko su neuronske mreže sposobne da prepoznaju nepoznate neispravnosti. Kad se kaže „nepoznate neispravnosti“, ovde se misli na nepoznate intenzitete razmatranih neispravnosti. U cilju prikupljanja podataka za testiranje neuronskih mreža na nivoima III i IV, izvedene su 2 grupe virtuelnih eksperimenata. Stanja usisnog sistema, koja su simulirana u tim eksperimentima, navedena su u Tab. 6.7 i Tab. 6.8.

Tab. 6.7 Pregled simuliranih nepoznatih neispravnosti – grupa 1

Stanje usisnog sistema	Oznaka	Napomena
Nema neispravnosti	Sve OK	
IAT senzor očitava za 17 stepeni nižu vrednost	IAT-17	Samo za slučaj bez protokomera
IAT senzor očitava za 12 stepeni nižu vrednost	IAT-12	Samo za slučaj bez protokomera
IAT senzor očitava za 12 stepeni višu vrednost	IAT+12	Samo za slučaj bez protokomera
IAT senzor očitava za 17 stepeni višu vrednost	IAT+17	Samo za slučaj bez protokomera
MAP senzor očitava za 12 kPa nižu vrednost	MAP-12	
MAP senzor očitava za 7 kPa nižu vrednost	MAP-7	
MAP senzor očitava za 7 kPa višu vrednost	MAP+7	
MAP senzor očitava za 12 kPa višu vrednost	MAP+12	
MAF senzor očitava 12 % nižu vrednost	MAF88%	Samo za slučaj sa masenim protokomerom
MAF senzor očitava 7 % nižu vrednost	MAF93%	Samo za slučaj sa masenim protokomerom
MAF senzor očitava 7 % višu vrednost	MAF107%	Samo za slučaj sa masenim protokomerom
MAF senzor očitava 12 % višu vrednost	MAF112%	Samo za slučaj sa masenim protokomerom
Ulaz „falš“-vazduha kroz otvor prečnika 1.7mm	FaV1.7mm	
Ulaz „falš“-vazduha kroz otvor prečnika 3.6mm	FaV3.6mm	
Ulaz „falš“-vazduha kroz otvor prečnika 5.4mm	FaV5.4mm	
EGR protok umanjen na 85%	EGR85%	
EGR protok umanjen na 65%	EGR65%	

Tab. 6.8 Pregled simuliranih nepoznatih neispravnosti – grupa 2

Stanje usisnog sistema	Oznaka	Napomena
Nema neispravnosti	Sve OK	
IAT senzor očitava za 13 stepeni nižu vrednost	IAT-13	Samo za slučaj bez protokomera
IAT senzor očitava za 6 stepeni nižu vrednost	IAT-6	Samo za slučaj bez protokomera
IAT senzor očitava za 6 stepeni višu vrednost	IAT+6	Samo za slučaj bez protokomera
IAT senzor očitava za 13 stepeni višu vrednost	IAT+13	Samo za slučaj bez protokomera
MAP senzor očitava za 9 kPa nižu vrednost	MAP-9	
MAP senzor očitava za 4 kPa nižu vrednost	MAP-4	
MAP senzor očitava za 4 kPa višu vrednost	MAP+4	
MAP senzor očitava za 9 kPa višu vrednost	MAP+9	
MAF senzor očitava 8 % nižu vrednost	MAF92%	Samo za slučaj sa masenim protokomerom
MAF senzor očitava 4 % nižu vrednost	MAF96%	Samo za slučaj sa masenim protokomerom
MAF senzor očitava 4 % višu vrednost	MAF104%	Samo za slučaj sa masenim protokomerom
MAF senzor očitava 8 % višu vrednost	MAF108%	Samo za slučaj sa masenim protokomerom
Ulaz „falš“-vazduha kroz otvor prečnika 1.4mm	FaV1.4mm	
Ulaz „falš“-vazduha kroz otvor prečnika 4.3mm	FaV4.3mm	
Ulaz „falš“-vazduha kroz otvor prečnika 5.7mm	FaV5.7mm	
EGR protok umanjen na 73%	EGR73%	
EGR protok umanjen na 57%	EGR57%	

### Nivo III

Osnovna karakteristika testiranja na nivou III su nepoznate neispravnosti, ali su te neispravnosti simulirane u poznatim baznim radnim tačkama. Konkretno, za potrebe testiranja svake neuronske mreže, proizvoljno su odabrane po dve radne tačke iz Tab. 6.4 na strani 128. U obe radne tačke izvršene su simulacije neispravnosti iz grupe 1 (Tab. 6.7) i grupe 2 (Tab. 6.8), što čini četiri virtuelna eksperimenta po jednoj neuronskoj mreži odnosno 32 virtuelna eksperimenta po dijagnostičkom modelu. U kojim baznim radnim tačkama su izvođeni virtuelni eksperimenti, može se videti u Tab. 6.9, koja prikazuje plan izvođenja virtuelnih eksperimenata na ovom nivou testiranja neuronskih mreža.

Tab. 6.9 Plan izvođenja virtuelnih eksperimenata - testiranje neuronskih mreža (Nivo III)

Rad motora u zatvorenoj sprezi sa $\lambda$ -sondom						Rad motora u otvorenoj sprezi sa $\lambda$ -sondom					
Neur. mreža	Redni broj eksper.	Grupa virtuelnih eksperim.	Stepen prenosa	Temp. okoline [°C]	Otvor leptira [%]	Neur. mreža	Red. broj eksper.	Grupa virtuelnih eksperim.	Stepen prenosa	Temp. okoline [°C]	Otvor leptira [%]
NM <sub><math>\lambda</math></sub> <sup>II</sup>	1	1	II	0	5	NM <sup>II</sup>	17	1	II	0	5
	2			10	6		18			10	6
	3	2		0	5		19	2		0	5
	4			10	6		20			10	6
NM <sub><math>\lambda</math></sub> <sup>III</sup>	5	1	III	20	7	NM <sup>III</sup>	21	1	III	20	7
	6			30	8		22			30	8
	7	2		20	7		23	2		20	7
	8			30	8		24			30	8
NM <sub><math>\lambda</math></sub> <sup>IV</sup>	9	1	IV	0	11	NM <sup>IV</sup>	25	1	IV	0	11
	10			10	13		26			10	13
	11	2		0	11		27	2		0	11
	12			10	13		28			10	13

Tab 6.9 (Nastavak)

NM <sub>λ</sub> <sup>V</sup>	13	1	V	20	13	NM <sup>V</sup>	29	1	V	20	13
	14			30	17		30			17	
	15	2		20	13		31	2		20	13
	16			30	17		32			30	17

#### Nivo IV

Četvrti nivo testiranja neuronskih mreža razlikuje se od trećeg po tome, što su podaci za ovaj nivo testiranja prikupljeni izvođenjem virtuelnih eksperimenata u nepoznatim baznim radnim tačkama. Za potrebe testiranja svake neuronske mreže, odabrane su po dve bazne radne tačke i to na taj način, da vrednosti temperature okoline i otvora leptira budu unutar odgovarajućih opsega, definisanih virtuelnim eksperimentima koji su izvedeni u cilju prikupljanja podataka za obuku neuronskih mreža (Tab. 6.4 na strani 128). U ovako odabranim radnim tačkama, simulirana su stanja usisnog sistema iz Tab. 6.7, izvođenjem virtuelnih eksperimenata grupe 1, a izvođenjem virtuelnih eksperimenata grupe 2, simulirana su stanja usisnog sistema iz Tab. 6.8. Ovo će biti jasnije, ako se pogleda plan virtuelnih eksperimenata za ovaj nivo testiranja, dat u Tab. 6.10.

Tab. 6.10 Plan izvođenja virtuelnih eksperimenata - testiranje neuronskih mreža (Nivo IV)

Rad motora u zatvorenoj sprezi sa λ-sondom						Rad motora u otvorenoj sprezi sa λ-sondom					
Neur. mreža	Redni broj eksper.	Grupa virtuelnih eksperim.	Stepen prenosa	Temp. okoline [°C]	Otvor leptira [%]	Neur. mreža	Red. broj. eksper.	Grupa virtuelnih eksperim.	Stepen prenosa	Temp. okoline [°C]	Otvor leptira [%]
NM <sub>λ</sub> <sup>II</sup>	1	1	II	12	5.3	NM <sup>II</sup>	17	1	II	12	5.3
	2			28	6.7		18			6.7	
	3	2		12	5.3		19	2		12	5.3
	4			28	6.7		20			28	6.7
NM <sub>λ</sub> <sup>III</sup>	5	1	III	12	7.4	NM <sup>III</sup>	21	1	III	12	7.4
	6			28	8.2		22			8.2	
	7	2		12	7.4		23	2		12	7.4
	8			28	8.2		24			28	8.2
NM <sub>λ</sub> <sup>IV</sup>	9	1	IV	12	10.5	NM <sup>IV</sup>	25	1	IV	12	10.5
	10			28	12.5		26			28	12.5
	11	2		12	10.5		27	2		12	10.5
	12			28	12.5		28			28	12.5
NM <sub>λ</sub> <sup>V</sup>	13	1	V	12	14	NM <sup>V</sup>	29	1	V	12	14
	14			28	18		30			28	18
	15	2		12	14		31	2		12	14
	16			28	18		32			28	18

Tabela pokazuje da ukupan broj eksperimenata izvedenih za potrebe testiranja jednog dijagnostičkog modela iznosi 32. Isti ovaj plan virtuelnih eksperimenata bio je osnova i za testiranje drugog dijagnostičkog modela.

#### Nivo V

Cilj testiranja na ovom nivou je da se ilustruje kako promene nekih spoljašnjih faktora, čije varijacije nisu uzete u obzir prilikom generisanja skupa podataka za obuku neuronskih mreža, utiču na performanse dijagnostičkog modela. Da bi se to postiglo, menjane su vrednosti nekih ulaznih podataka u simulatoru neispravnosti u odnosu na njihove vrednosti koje su korišćene pri obuci neuronskih mreža. Konkretno, uticajni faktori koji su uzeti u obzir pri testiranju su:

- promena mase vozila,
- vetar u pravcu kretanja vozila,

- promena atmosferskog pritiska i
- promena uzdužnog nagiba puta.

Masa vozila je varirana na 900 kg i 1100 kg, u odnosu na inicijalnih 1000 kg. Što se tiče vetra u pravcu kretanja vozila, on nije uzet u obzir tokom obuke neuronskih mreža, što znači da je inicijalna brzina vetra bila 0. Na ovom nivou, izvršena su testiranja sa brzinom vetra 15 km/h, suprotno smeru kretanja vozila i 15 km/h u smeru kretanja vozila. Inicijalni atmosferski pritisak od 1 bar, variran je na vrednosti 0.98 bar i 1.02 bar. Prilikom razvoja dijagnostičkog modela uzdužni nagib puta je imao vrednost 0, a na ovom testiranju njegove vrednosti su varirane sa -1% i 1%. Svaka od ovih promena pojedinačno je primenjivana u virtuelnim eksperimentima u kojima su simulirane nepoznate neispravnosti u nepoznatim baznim radnim tačkama motora. Drugim rečima, testiranje na nivou V na neki način predstavlja pooštavanje uslova testiranja na nivou IV.

Cilj testiranja na nivou V je samo da se ilustruje robusnost dijagnostičkog modela, tj. njegova osetljivost na realne promene nekih faktora koji mogu da utiču na ulazne veličine koje su korišćene za obuku neuronskih mreža. U tom smislu, testiranja su izvršena samo na dijagnostičkom modelu usisnog sistema bez protokomera vazduha i to samo u III i IV stepenu prenosa. Pri tome, uzete su u obzir samo nepoznate neispravnosti grupe 1 iz Tab. 6.7. Ova ograničenja su uvedena da bi se broj virtuelnih eksperimenata sveo na neku razumnu meru, uz pretpostavku da će i to biti dovoljno da se stekne uvid u robusnost razvijenog dijagnostičkog modela.

U Tab. 6.11 daje se plan izvođenja virtuelnih eksperimenata za V nivo testiranja.

Tab. 6.11 Plan izvođenja virtuelnih eksperimenata - testiranje neuronskih mreža (Nivo V)

Rad motora u zatvorenoj sprezi sa $\lambda$ -sondom									
Osnovni podaci i bazna radna tačka						Parametri koji se variraju			
Neur. mreža	Red. br. eksper.	Grupa eksperim.	Stepen prenosa	Temp. okoline [°C]	Otvor leptira [%]	Masa vozila [kg]	Brzina vetra [km/h]	Atmosferski pritisak [bar]	Nagib puta [%]
NM $_{\lambda}^{III}$	1	1	III	12	7.4	900	0	1	0
	2					1100	0	1	0
	3					1000	+15	1	0
	4					1000	-15	1	0
	5					1000	0	0.98	0
	6					1000	0	1.02	0
	7					1000	0	1	-1
	8					1000	0	1	1
NM $_{\lambda}^V$	9		IV	12	10.5	900	0	1	0
	10					1100	0	1	0
	11					1000	+15	1	0
	12					1000	-15	1	0
	13					1000	0	0.98	0
	14					1000	0	1.02	0
	15					1000	0	1	-1
	16					1000	0	1	1
Rad motora u otvorenoj sprezi sa $\lambda$ -sondom									
NM $_{\lambda}^{III}$	17	1	III	12	7.4	900	0	1	0
	18					1100	0	1	0
	19					1000	+15	1	0
	20					1000	-15	1	0
	21					1000	0	0.98	0
	22					1000	0	1.02	0
	23					1000	0	1	-1
	24					1000	0	1	1



Tab 6.11 (Nastavak)

NM <sub>λ</sub> <sup>y</sup>	25	1	IV	12	10.5	900	0	1	0
	26					1100	0	1	0
	27					1000	+15	1	0
	28					1000	-15	1	0
	29					1000	0	0.98	0
	30					1000	0	1.02	0
	31					1000	0	1	-1
	32					1000	0	1	1

### 6.3.6 Računarski programi za praktičnu realizaciju dijagnostičkog modela

Ranije je napomenuto da je praktična realizacija strukture neuronskih mreža dijagnostičkog modela i njihove obuke izvedena pomoću namenski napisanog računarskog programa, pod nazivom *kreirajNetStrukturu*. U okviru istog programa, realizuje se i jedno testiranje obučene mreže. Izvorni kod ovog programa dat je u prilogu P3.3, a ovde se prikazuje njegov pseudokod:

**Program** *kreirajNetStrukturu*

**Počni**

**Učitaj** potrebne parametre za kreiranje neuronske mreže

**Učitaj** strukturu za obuku iz Matlab-fajla

**Izdvoj** ulaznu matricu za obuku i matricu željenih izlaza iz strukture za obuku

**Kreiraj** Network objekat<sup>30</sup> pozivom Matlab funkcije za kreiranje BP neuronske mreže<sup>31</sup> nad ulaznom matricom za obuku i matricom željenih izlaza

**Dodeli** vrednosti parametrima za obuku neuronske mreže u Network objektu

**Pozovi** Matlabovu funkciju za obuku neuronske mreže

**Formiraj** matricu za testiranje pozivom funkcije *generisiTestStrukturu*

**Testiraj** neuronsku mrežu iz Network objekta podacima iz matrice za testiranje

**Objedini** rezultate testiranja, Network objekat i pripadajuće parametre u jednu Matlabovu strukturu (NN struktura)

**Generiši** naziv Matlab fajla

**Snimi** NN strukturu u Matlab fajl

**Iscrtaj** rezultate testiranja pozivom programa *iscrtajNNizlaz*

**Završi**

U okviru programa *kreirajNetStrukturu* pozivaju se funkcija *generisiTestStrukturu* i program *iscrtajNNizlaz*.

Funkcija *generisiTestStrukturu* je veoma jednostavna, jer samo omogućava preuzimanje podataka iz odabranog Excel-fajla, od njih formira matricu prilagođenu za testiranje neuronske mreže koju, zajedno sa imenom odabranog Excel-fajla, grupiše u jednu Matlab-ovu strukturu. Izvorni kod ove funkcije dat je u prilogu P3.4.

Program *iscrtajNNizlaz* obezbeđuje vizuelni prikaz rezultata testiranja neuronske mreže (sa odgovarajućim oznakama svih razmatranih stanja usisnog sistema) i snima iscrtani

<sup>30</sup> Matlab objekat koji sadrži neuronsku mrežu. Reč „Network“ je upotrebljena zbog engleskog naziva za mrežu.

<sup>31</sup> Pozivom funkcije za kreiranje neuronske mreže automatski se inicijalizuju težinski koeficijenti i vrši se normalizacija podataka i njihova podela na podskupove za obuku, validaciju i testiranje.

vizuelni prikaz u fajl sa automatski generisanim imenom. Na taj način se omogućava veoma brzi uvid u sposobnost neuronske mreže da dijagnostikuje neispravnosti u usisnom sistemu, bez potrebe za izračunavanjem nekih statističkih parametara da bi se ocenio njen kvalitet. Zbog toga je ovaj program veoma koristan, a još više dobija na značaju, ako se pomene da je u toku izvođenja ovog istraživanja (zajedno sa preliminarnim istraživanjima) trebalo iscrtati više od 2000 dijagrama. Izvorni kod programa *iscrtajNNizlaz*, zajedno sa pripadajućim pomoćnim programima, prikazan je u prilogu P4.1.

Za testiranje već generisanih neuronskih mreža koristi se namenski napisani program *testirajNetStrukturu*, čiji je izvorni kod dat u prilogu P3.5, a pseudokod je prikazan ovde:

### Program *testirajNetStrukturu*

#### Počni

**Učitaj** potrebne parametre za prikaz rezultata testiranja

**Učitaj NN strukturu** iz Matlab-fajla

**Izdvoj Network objekat** iz učitane NN strukture

**Formiraj matricu za testiranje** pozivom funkcije *generisiTestStrukturu*

**Testiraj** neuronsku mrežu iz *Network objekta* podacima iz *matrice za testiranje*

**Isfiltriraj izlaz** testirane neuronske mreže pozivom funkcije *preradiRedukujNNizlaz*

**Formiraj testiranu NN strukturu** od *Network objekta*, *matrice za testiranje*, *rezultata testiranja* i *filtriranih rezultata testiranja*

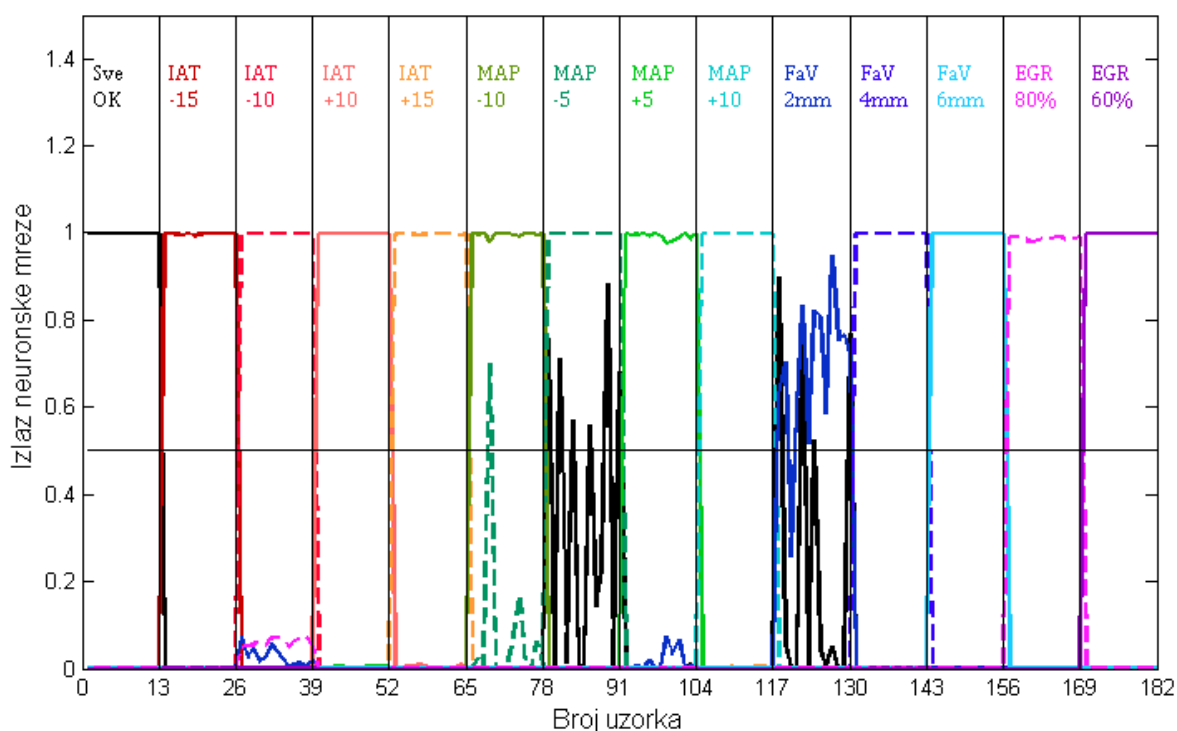
**Generiši naziv novog Matlab fajla**

**Snimi testiranu NN strukturu** u novi Matlab fajl

**Prikaži rezultate testiranja** pozivom programa *iscrtajNNizlaz*

#### Završi

Na kraju izvršenja programa *testirajNetStrukturu*, dobija se vizuelni prikaz rezultata testiranja neuronske mreže, čiji je jedan primer, radi ilustracije, pokazan na Sl. 6.5.



Sl. 6.5 Primer vizuelnog prikaza rezultata testiranja neuronske mreže

Dijagram na *Sl. 6.5* prikazuje 14 različito obojenih linija, od kojih svaka povezuje 182 diskretne izlazne vrednosti po jednog od 14 izlaznih neurona, za 182 ulazna uzorka iz skupa podataka za testiranje. Od ovih 182 uzorka, po 13 ih je snimljeno pri simulaciji svakog stanja usisnog sistema motora. Zato je u dijagramu na svakih 13 uzoraka ucrtana po jedna vertikalna linija, čime je on podeljen na 14 oblasti. Svaka oblast je u svom gornjem delu obeležena prepoznatljivom oznakom stanja, kako bi se znalo kojem stanju usisnog sistema odgovara svaki ulazni uzorak. Pomoću ovih oznaka može se raspoznati i koja linija na dijagramu predstavlja odziv kojeg izlaznog neurona. Naime, ako su neka linija i oznaka stanja u istoj boji, to znači da se one odnose na isti izlazni neuron. Na primer, crnom linijom na dijagramu prikazan je odziv izlaznog neurona koji reprezentuje stanje usisnog sistema bez neispravnosti, jer je oznaka tog stanja (SveOK) takođe crne boje.

Na dijagramu je ucrtana i horizontalna linija praga detektovanja nekog stanja, za koji je ranije usvojena vrednost 0.5. Na osnovu te vrednosti, posmatranjem dijagrama se vrlo lako može doneti zaključak o uspešnosti detektovanja stanja usisnog sistema od strane konkretne neuronske mreže. Naime, neko stanje je uspešno detektovano, ako se na dijagramu, u oblasti koja odgovara tom stanju, samo linija odziva odgovarajućeg neurona nalazi iznad linije praga, dok su linije odziva svih ostalih neurona ispod linije praga. Međutim, ovo ne mora tako kruto da se posmatra, s obzirom da različiti izlazni neuroni mogu da se odnose na različite intenzitete iste vrste neispravnosti. O ovome će biti više reči u komentarima konkretnih rezultata testiranja, u odeljku 6.3.7.

Dijagram na *Sl. 6.5* pokazuje još jedno interesantno svojstvo izlaznih vrednosti neuronske mreže. Primećuje se da linije odziva nekih izlaznih neurona imaju izražen oscilatorni karakter. Konkretno, na datom dijagramu ova konstatacija važi za izlazne neurone označene sa „SveOK“, „MAP-5“, i „FaV2mm“, ali na nekim drugim dijagramima to može biti slučaj sa bilo kojim izlaznim neuronom. Ovakvo ponašanje linije odziva, posledica je pulsacija vrednosti ulaznih veličina neuronske mreže, pre svega broja obrtaja motora, pritiska u usisnom kolektoru i masenog protoka vazduha. Oscilatorni karakter odziva izlaznih neurona ima negativan uticaj na sposobnost neuronske mreže da prepozna trenutno stanje usisnog sistema, jer zbog velikog broja pikova, linija odziva često prelazi preko linije praga u oba smera. To dovodi do povećanog broja lažnih alarma ili do izostanka identifikacije nekog stanja.

Da bi se ove greške svele na što manju meru i da bi se poboljšale performanse dijagnostičkog modela, odlučeno je da se trenutno stanje usisnog sistema ne identifikuje na osnovu izlaza neuronske mreže samo u tom trenutku, nego da se uzme u obzir i nekoliko prethodnih uzastopnih izlaza neuronske mreže. Ova ideja je realizovana pomoću namenski napisane program-funkcije *preradiRedukujNNizlaz*, koja kao ulazne parametre prihvata originalnu matricu izlaza neuronske mreže i celobrojni parametar  $z$  koji označava broj uzastopnih izlaza koji se žele uzeti u obzir pri dijagnostikovanju stanja. Kao izlaz funkcije *preradiRedukujNNizlaz* dobija se prerađena matrica izlaza neuronske mreže. Izvorni kod ove funkcije dat je u prilogu P3.6, a pseudokod se prikazuje ovde:

**Program** *preradiRedukujNNizlaz*

**Počni**

**Učitaj** *matricu izlaza neuronske mreže*

**Učitaj** parametar  $z$

**Odredi** *ukupan broj stanja usisnog sistema*

**Odredi** *broj uzoraka po stanju*

**Postavi** *brojač stanja na 1*

**Dok** *brojač stanja ne premaši ukupan broj stanja, radi*

**Izdvoj** matricu tekućeg stanja iz matrice izlaza neuronske mreže

**Postavi** brojač izlaznih neurona na 1

**Dok** brojač izlaznih neurona ne premaši ukupan broj izlaznih neurona, **radi**

**Postavi** brojač prerađenih uzoraka na  $z$

**Dok** brojač prerađenih uzoraka ne premaši broj uzoraka po stanju, **radi**

**Izračunaj** srednju vrednost prethodnih  $z$  odziva tekućeg izlaznog neurona

**Upiši** izračunatu srednju vrednost u matricu tekućeg stanja na poziciju (brojač izlaznih neurona, brojač prerađenih uzoraka)

**Povećaj** brojač prerađenih uzoraka za 1

**Povećaj** brojač izlaznih neurona za 1

**Formiraj** prerađenu matricu tekućeg stanja brisanjem prvih ( $z-1$ ) kolona matrice tekućeg stanja

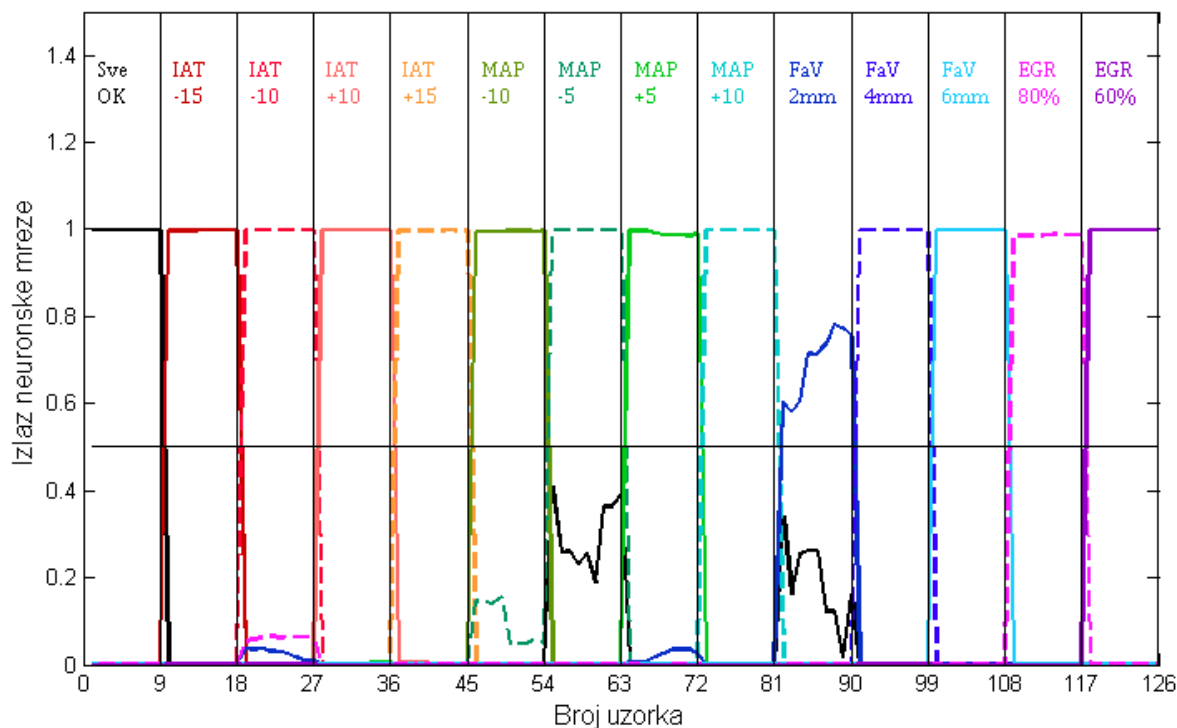
**Preradi** matricu izlaza neuronske mreže zamenom matrice tekućeg stanja prerađenom matricom tekućeg stanja

**Povećaj** brojač stanja za 1

**Vrati** prerađenu matricu izlaza neuronske mreže

**Završi**

Ova funkcija se poziva u okviru ranije pomenutog programa *testirajNetStrukturu*, a efekat koji se pomoću nje dobija, ilustrovan je na Sl. 6.6. Dijagram na Sl. 6.6 nastao je primenom funkcije *preradiRedukujNNizlaz* na matricu izlaza neuronske mreže ilustrovanu na Sl. 6.5, pri čemu je za parametar  $z$  usvojena vrednost 5. To znači da je trenutna vrednost svakog prerađenog izlaza neuronske mreže dobijena uprosečavanjem trenutne i još četiri prethodne vrednosti odgovarajućeg neprerađenog izlaza neuronske mreže. Zbog toga je broj uzoraka po jednom stanju na Sl. 6.6 za četiri manji od broja uzoraka po stanju na Sl. 6.5.



Sl. 6.6 Primer vizuelnog prikaza prerađenih izlaza neuronske mreže

Očigledno je da je, sa aspekta prepoznavanja stanja usisnog sistema, dijagram na *Sl. 6.6* mnogo jasniji od svog prethodnika sa *Sl. 6.5*, jer su oscilatorni pikovi neutralisani primenom funkcije *preradiRedukujNNizlaz*. Primećuje se da su sva stanja usisnog sistema u potpunosti uspešno identifikovana, s obzirom da za svih 14 stanja, samo odgovarajući neuroni imaju odziv iznad 0.5, a odzivi svih ostalih izlaznih neurona su ispod te vrednosti.

Treba još napomenuti da funkcija *preradiRedukujNNizlaz* nije predviđena da radi u realnom vremenu, nego je osmišljena kao alat koji bi trebalo da pokaže da li i u kojoj meri se dodatnom obradom podataka mogu poboljšati performanse razvijenog dijagnostičkog modela.

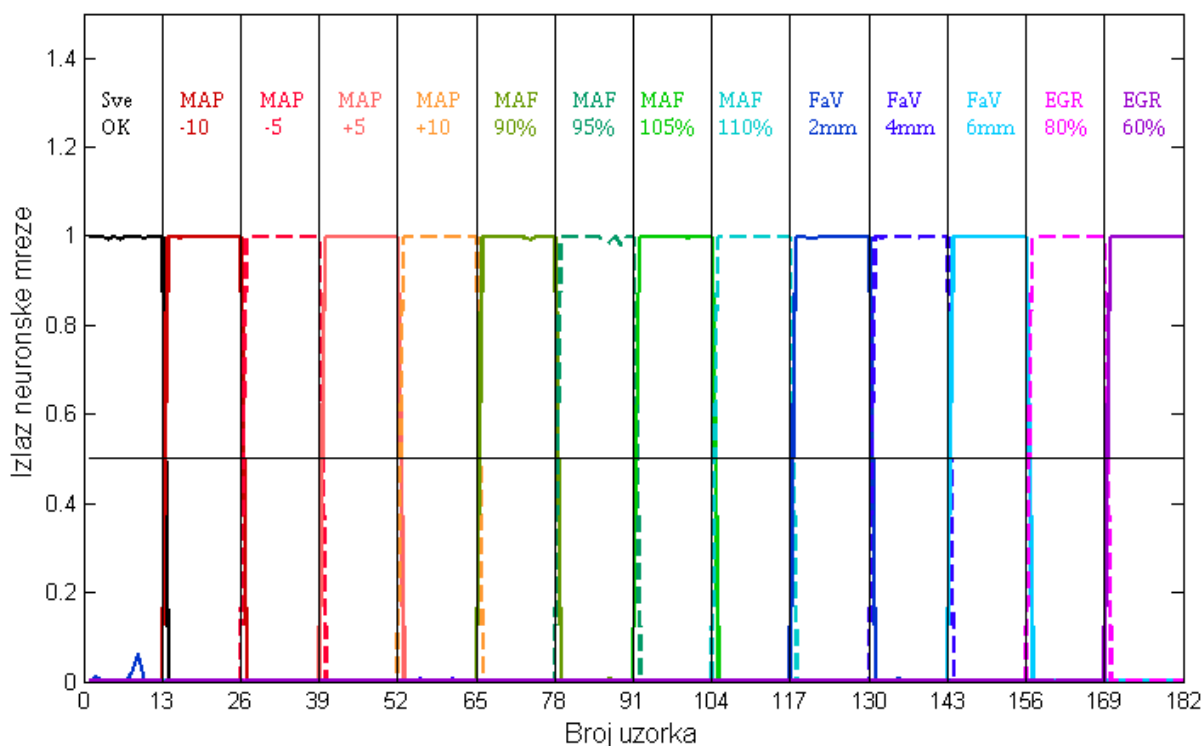
### 6.3.7 Rezultati testiranja dijagnostičkog modela i diskusija

S obzirom na to da su razvijena dva dijagnostička modela sa po osam neuronskih mreža, kao i na činjenicu da je testiranje izvedeno na 5 različitih nivoa, količina rezultata dobijenih testiranjem je prilično velika. Grafički prikaz svih dobijenih rezultata čini 256 dijagrama sličnih onome na *Sl. 6.6*. Zbog ovako velikog broja dijagrama, rezultati testiranja su u celosti dati na priloženom CD-ROM-u, a ovde se prikazuje deo tih rezultata, odabran tako, da se stekne uvid u mogućnosti razvijenih dijagnostičkih modela.

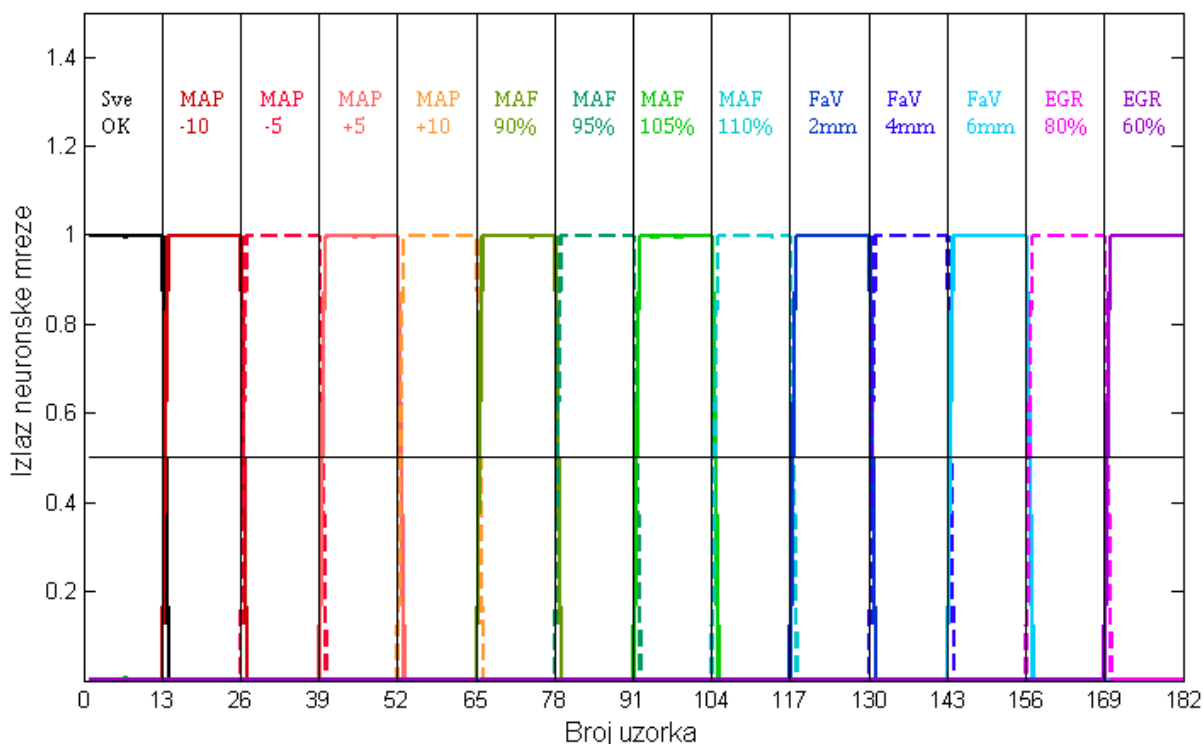
Rezultati će biti prikazani po nivoima testiranja, od prvog do petog, a u okviru svakog nivoa biće dat odgovarajući broj reprezentativnih primera za oba dijagnostička modela (sa masenim protokomerom i bez protokomera vazduha). Pri tome će biti obuhvaćeni slučajevi rada motora u otvorenoj i zatvorenoj sprezi sa  $\lambda$ -sondom, u četiri razmatrana stepena prenosa.

#### **Nivo I – poznata stanja u poznatim baznim radnim tačkama**

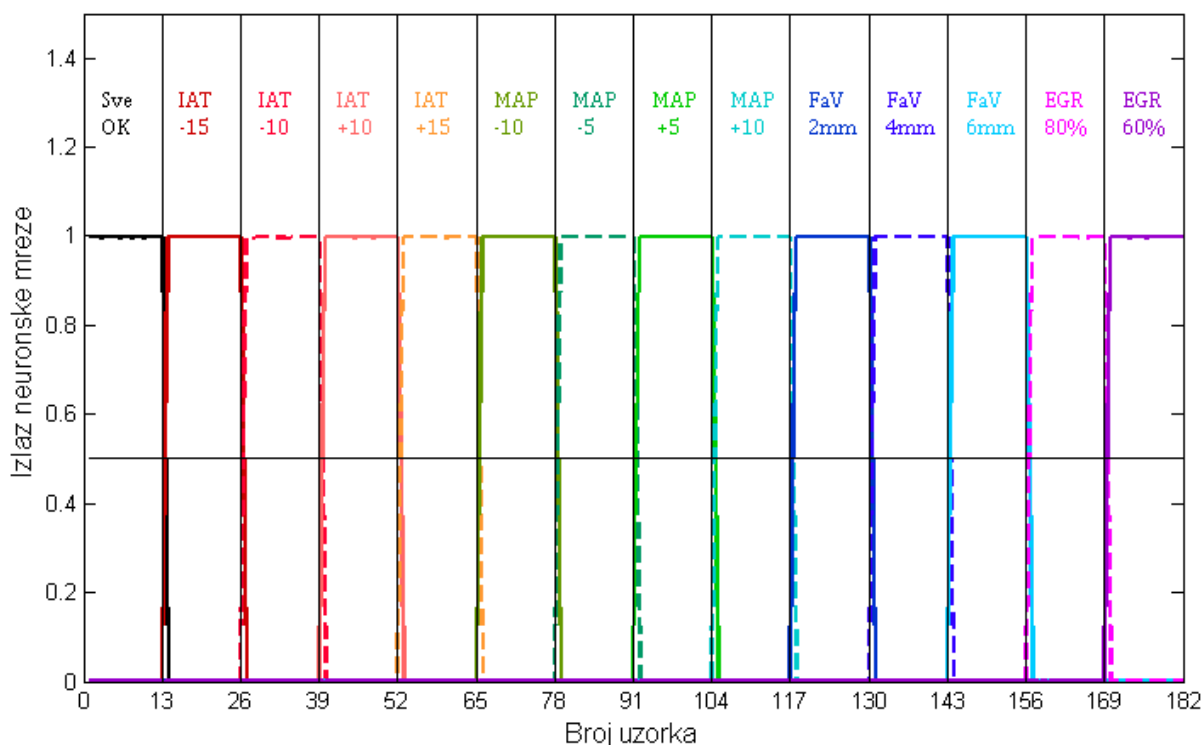
Rezultati testiranja na ovom nivou su toliko dobri, da su svi dobijeni dijagrami u okviru jednog dijagnostičkog modela gotovo identični, bez obzira na to, o kojoj se od osam neuronskih mreža radi i bez obzira na baznu radnu tačku u kojoj je sprovedeno testiranje. Zbog toga se, kao ilustracija za svaki dijagnostički model, daju samo po dva dijagrama, pri čemu su obuhvaćena sva četiri razmatrana stepena prenosa. Na *Sl. 6.7* i *Sl. 6.8* prikazani su odabrani primeri rezultata testiranja dijagnostičkog modela usisnog sistema sa masenim protokomerom, za slučajeve zatvorene i otvorene sprege sa  $\lambda$ -sondom, respektivno. Slično tome, na *Sl. 6.9* i *Sl. 6.10* dati su odabrani primeri rezultata testiranja dijagnostičkog modela usisnog sistema bez protokomera, za slučajeve zatvorene i otvorene sprege sa  $\lambda$ -sondom, respektivno.



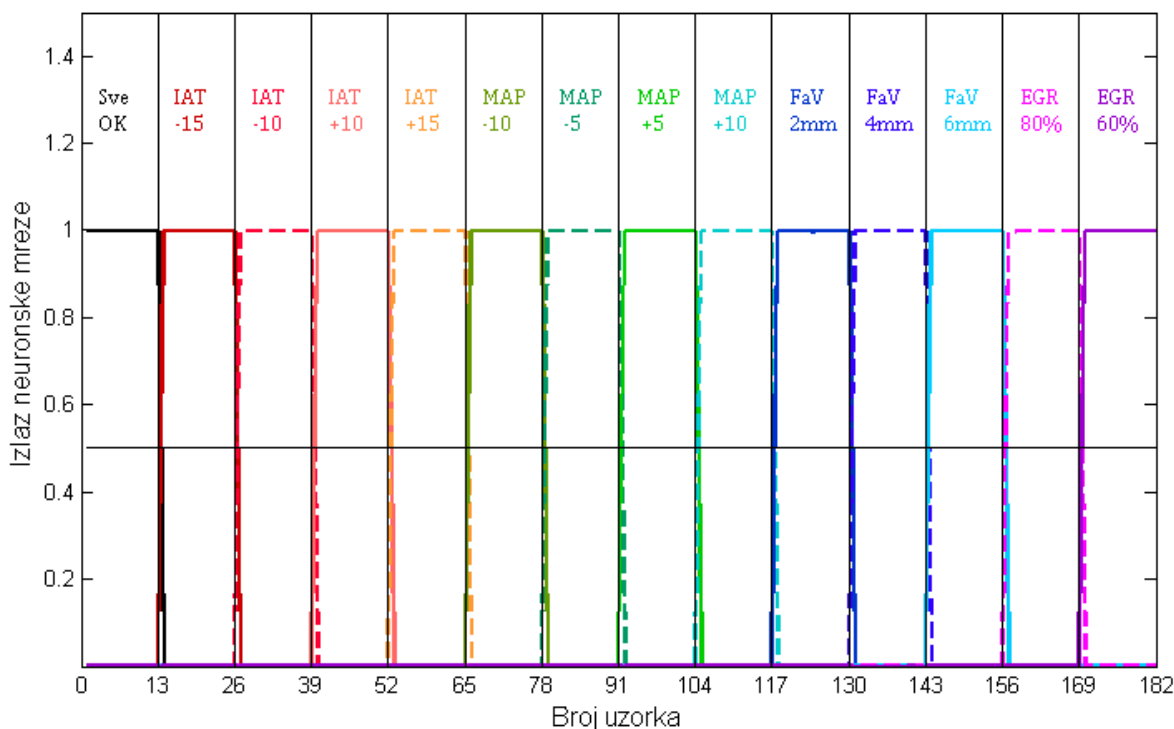
Sl. 6.7 Rezultati testiranja neuronske mreže – motor sa masenim protokomerom vazduha, zatvorena sprega sa  $\lambda$ -sondom, V stepen prenosa, otvor leptira 17%, temperatura okoline 30°C, greška  $1.7783 \cdot 10^{-7}$



Sl. 6.8 Rezultati testiranja neuronske mreže – motor sa masenim protokomerom vazduha, otvorena sprega sa  $\lambda$ -sondom, III stepen prenosa, otvor leptira 7%, temperatura okoline 20°C, greška  $2.0731 \cdot 10^{-7}$



Sl. 6.9 Rezultati testiranja neuronske mreže – motor bez protokomera vazduha, zatvorena sprega sa  $\lambda$ -sondom, IV stepen prenosa, otvor leptira 11%, temperatura okoline 0°C, greška  $3.4562 \cdot 10^{-7}$



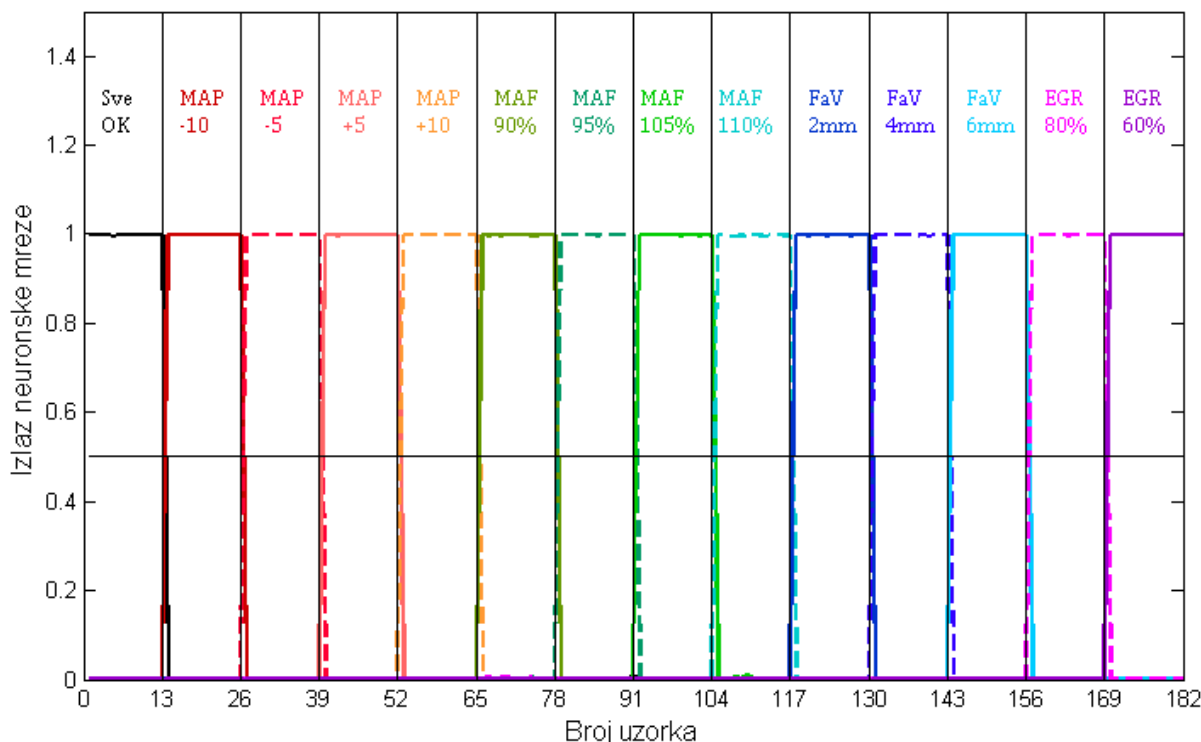
Sl. 6.10 Rezultati testiranja neuronske mreže – motor bez protokomera vazduha, otvorena sprega sa  $\lambda$ -sondom, II stepen prenosa, otvor leptira 6%, temperatura okoline 10°C, greška  $1.866 \cdot 10^{-7}$

Sve četiri slike pokazuju da dijagnostički modeli u potpunosti prepoznaju sva posmatrana stanja usisnog sistema. Pri tome, odzivi odgovarajućih izlaznih neurona ne samo da su iznad vrednosti praga, 0.5, nego su za sve ulazne uzorke ili jednaki jedinici ili veoma bliski toj vrednosti. Još treba napomenuti da se, na ovom nivou testiranja,

prepoznavanje stanja može uspješno izvesti na bazi trenutnih vrednosti izlaznih neurona, bez potrebe da se koriste njihove prethodne izlazne vrednosti. Ovakvi rezultati na ovom nivou testiranja su očekivani, imajući u vidu da je testiranje sprovedeno u poznatim baznim radnim tačkama, tj. da su podaci, koji su korišćeni za testiranje, veoma slični podacima korišćenim za obuku neuronskih mreža.

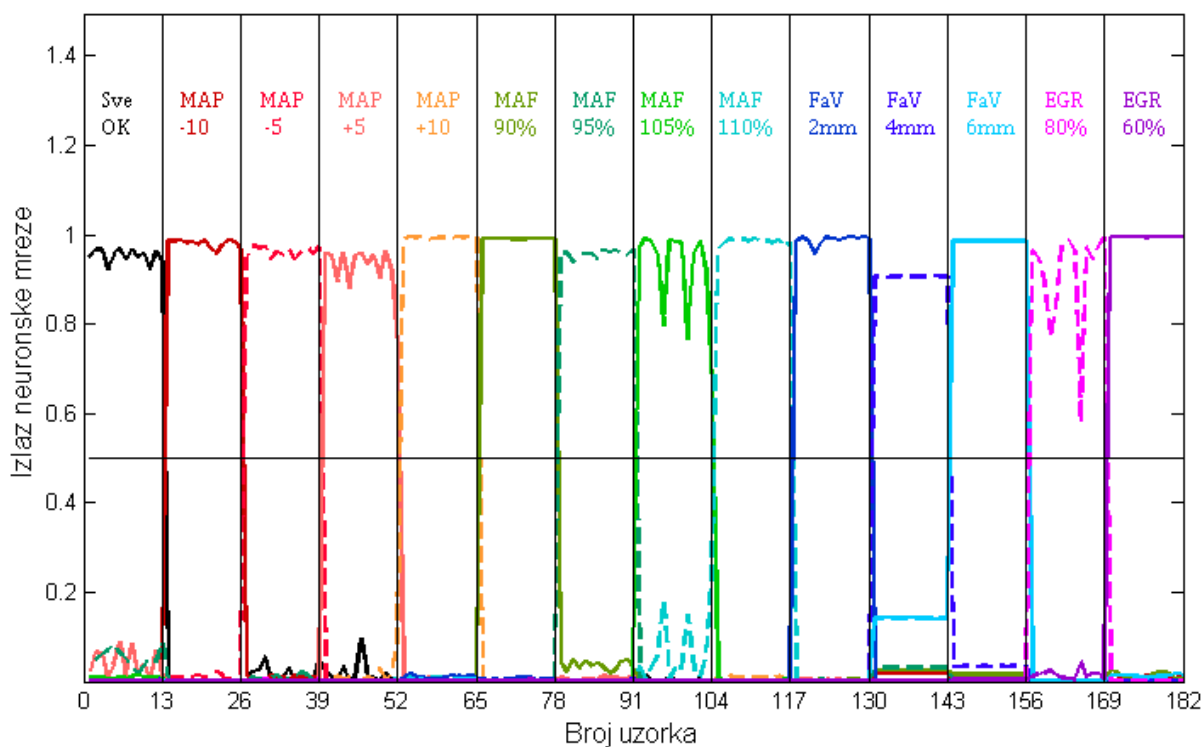
### Nivo II - poznata stanja u nepoznatim baznim radnim tačkama

Prilikom testiranja na ovom nivou takođe su dobijeni izuzetno dobri rezultati. Dijagrami dobijeni u svih 48 virtuelnih eksperimenata (ili 96, računajući oba dijagnostička modela), definisanih u *Tab. 6.6*, veoma su slični i jedni drugima, a ne razlikuju se upadljivo ni od dijagrama sa *Sl. 6.7 – Sl. 6.10*. Zato se i za ovaj nivo testiranja daju samo po dva dijagrama za svaki dijagnostički model, kao ilustracija. Pri odabiru dijagrama koji će ovde biti prikazani, nastojalo se da to budu oni, koji se odnose na bazne radne tačke kod kojih se i otvor leptira i temperatura okoline razlikuju u odnosu na bazne radne tačke korišćene pri obuci neuronskih mreža. Takođe, nastojalo se i da u odabranim dijagramima budu obuhvaćena sva četiri razmatrana stepena prenosa, kao i da bude prikazan dijagram koji reprezentuje najlošije rezultate. Na *Sl. 6.11* i *Sl. 6.12* prikazani su odabrani primeri rezultata testiranja dijagnostičkog modela usisnog sistema sa masenim protokomerom, za slučajeve zatvorene i otvorene sprege sa  $\lambda$ -sondom, respektivno. Slično tome, na *Sl. 6.13* i *Sl. 6.14* dati su odabrani primeri rezultata testiranja dijagnostičkog modela usisnog sistema bez protokomera, za slučajeve zatvorene i otvorene sprege sa  $\lambda$ -sondom, respektivno. Dijagrami na sve četiri slike pokazuju da dijagnostički modeli u potpunosti prepoznaju sva posmatrana stanja usisnog sistema, jer su linije odziva odgovarajućih izlaznih neurona iznad linije praga (0.5).

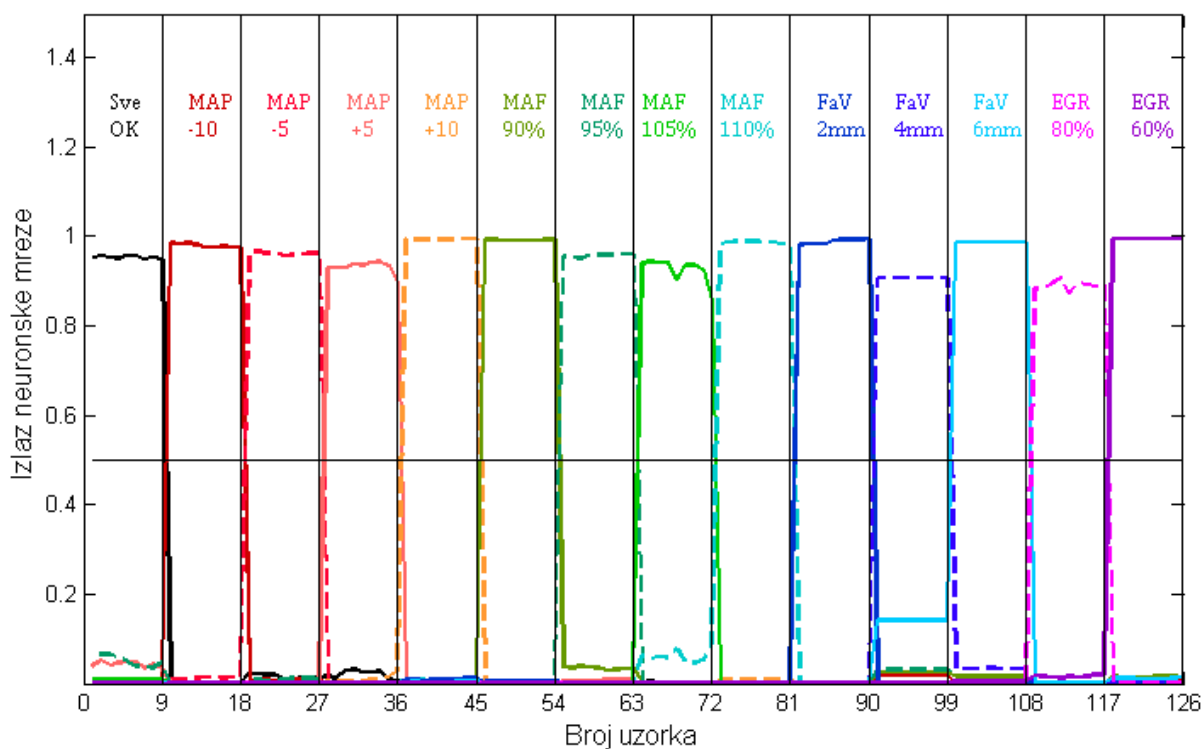


*Sl. 6.11* Rezultati testiranja neuronske mreže – motor sa masenim protokomerom vazduha, zatvorena sprega sa  $\lambda$ -sondom, IV stepen prenosa, otvor leptira 8.8%, temperatura okoline 0°C, greška  $1.8881 \cdot 10^{-7}$



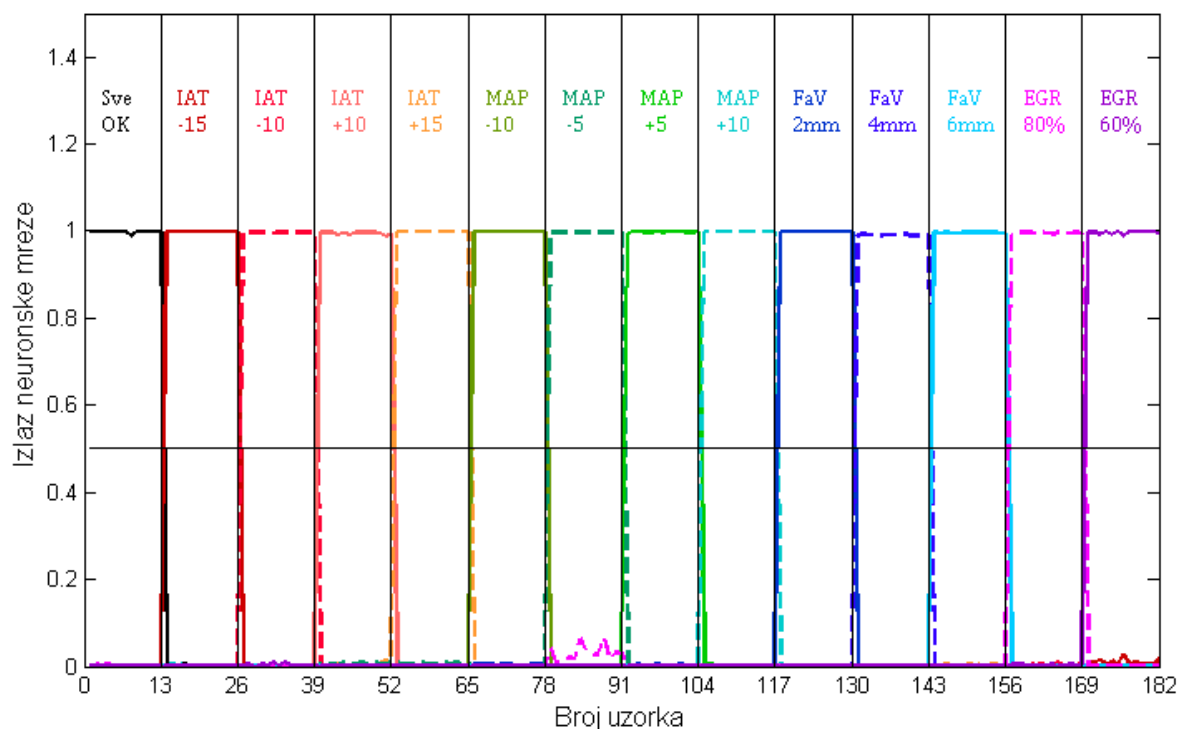


(a)

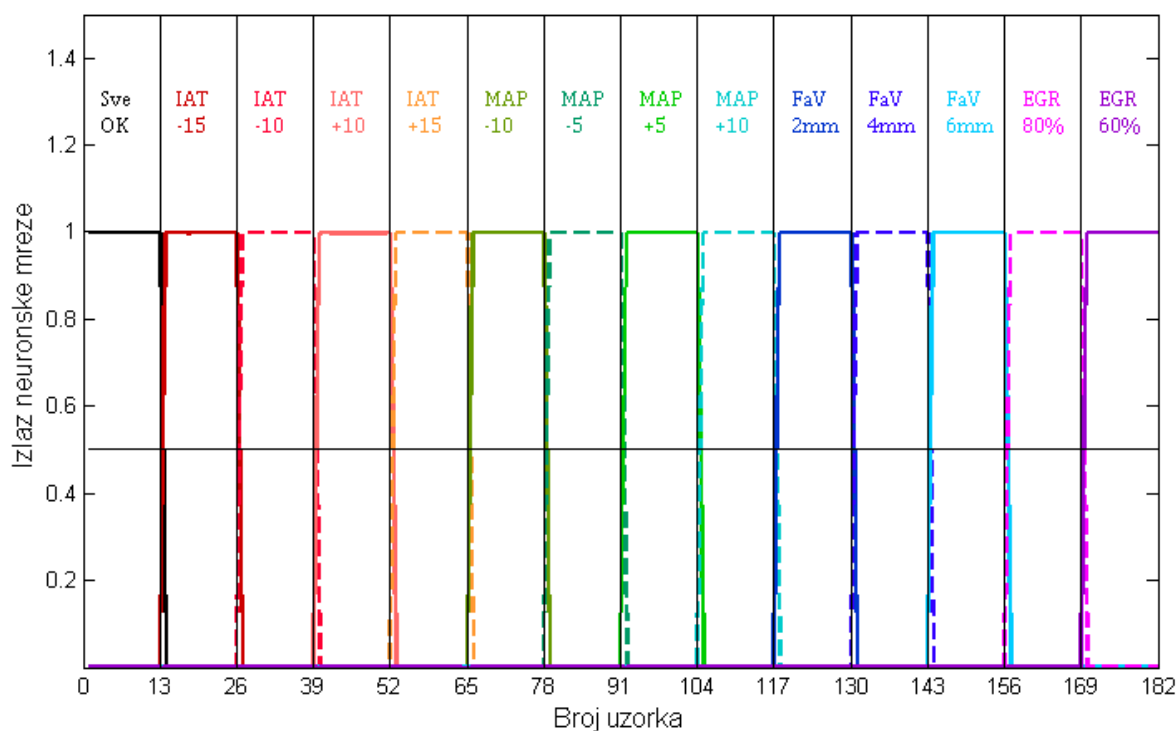


(b)

Sl. 6.12 Rezultati testiranja neuronske mreže – motor sa masenim protokomerom vazduha, otvorena sprega sa  $\lambda$ -sondom, II stepen prenosa, otvor leptira 7.2%, temperatura okoline 28°C, greška 0.0018526: (a) na bazi trenutnih vrednosti; (b) na bazi uprosечenih vrednosti



Sl. 6.13 Rezultati testiranja neuronske mreže – motor bez protokomera vazduha, zatvorena sprega sa  $\lambda$ -sondom, V stepen prenosa, otvor leptira 12%, temperatura okoline  $-5^{\circ}\text{C}$ , greška  $6.8749 \cdot 10^{-6}$



Sl. 6.14 Rezultati testiranja neuronske mreže – motor bez protokomera vazduha, otvorena sprega sa  $\lambda$ -sondom, III stepen prenosa, otvor leptira 7.3%, temperatura okoline  $-5^{\circ}\text{C}$ , greška  $3.1834 \cdot 10^{-7}$

Međutim, treba reći nekoliko reči o Sl. 6.12. Reč je o slučaju motora sa masenim protokomerom i režimu rada u otvorenoj sprezi sa  $\lambda$ -sondom. U ovom slučaju, pri obuci neuronske mreže dobijena je osetno veća minimalna vrednost funkcije greške (0.0018526), nego u ostala tri slučaja, gde funkcija greška ima vrednosti reda veličine  $10^{-6}$  ili  $10^{-7}$ .

Izvršeno je nekoliko desetina pokušaja obuke ove neuronske mreže sa novim, slučajno generisanim inicijalnim težinskim koeficijentima, ali nije dobijena manja vrednost funkcije greške od navedene. Razlog ovome je verovatno nemogućnost motora da radi pri relativno malim otvorima leptira u prisustvu neispravnosti „FaV4mm“ i „FaV6mm“, jer nema povratne sprege da obogati tako nastalu jako siromašnu smešu. To se može tretirati kao nepravilnost u skupu podataka za obuku, koja onemogućava postizanje nižih vrednosti funkcije greške. Na *Sl. 6.12* prikazani su rezultati istog testiranja, ali su u slučaju pod (a) ovi rezultati neobrađeni, a u slučaju pod (b) su obrađeni pomoću ranije opisane program-funkcije *preradiRedukujNNizlaz*. Prikazane su obe verzije kako bi se stekao malo bolji uvid u odziv neuronske mreže u ovom specifičnom slučaju.

Na osnovu priloženih dijagrama, primećuje se da se na ovom nivou testiranja, čak i u slučaju relativno velike greške, prepoznavanje svih stanja može u potpunosti izvesti na bazi trenutnih vrednosti izlaznih neurona, bez potrebe da se koriste njihove prethodne izlazne vrednosti.

### **Nivo III - nepoznata stanja u poznatim baznim radnim tačkama**

Testiranje neuronskih mreža na trećem nivou izvršeno je u po 32 bazne radne tačke za oba dijagnostička modela. Pri tome su dobijena 64 dijagrama rezultata testiranja, a ovde će biti dati samo neki od njih, odabrani tako da za oba dijagnostička modela budu obuhvaćeni i jedan i drugi  $\lambda$ -režim, sva četiri stepena prenosa i sva nepoznata stanja usisnog sistema. Dijagrami na *Sl. 6.15 - Sl. 6.18* odnose se na dijagnostički model usisnog sistema sa masenim protokomerom vazduha, a dijagrami na *Sl. 6.19 - Sl. 6.22* odnose se na slučaj bez protokomera vazduha.

Identifikacija nepoznatih stanja usisnog sistema je teži problem od identifikacije poznatih stanja, pa je neophodno izvršiti obradu odziva izlaznih neurona pomoću program-funkcije *preradiRedukujNNizlaz*, da bi se poboljšale performanse razvijenih dijagnostičkih modela. Zbog toga se na *Sl. 6.15 - Sl. 6.22* prikazuju samo obrađeni odzivi, sa izuzetkom *Sl. 6.15*, gde se, radi ilustracije, prikazuju i neobrađena i obrađena verzija izlaza neuronske mreže.

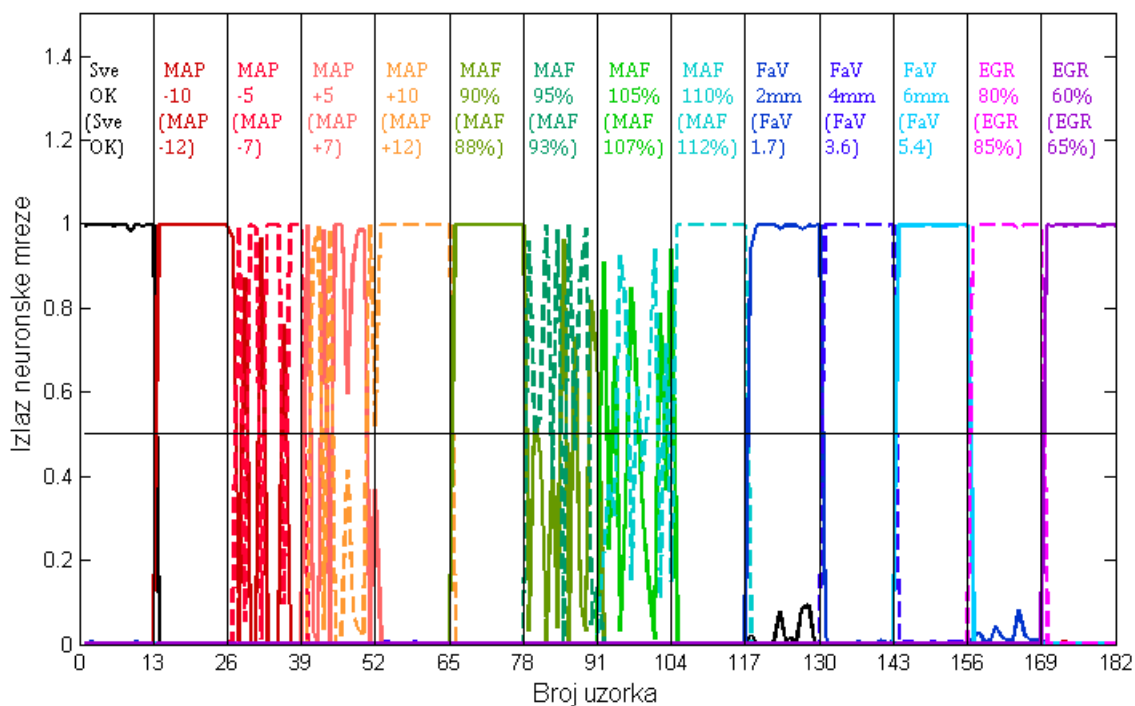
Treba napomenuti da se oznake na dijagramima koji slede, razlikuju od oznaka na dijagramima koji su se odnosili na I i II nivo testiranja, u smislu da je na predstojećim dijagramima, ispod oznaka poznatih od ranije, dodat još jedan red sličnih oznaka, koje su smeštene u zagradama. Značenje ovih oznaka je sledeće:

- Oznake u prvom redu, koje nisu u zagradama, odnose se na izlazne neurone obučene neuronske mreže.
- Oznake u drugom redu, koje su u zagradama, odnose se na stanja usisnog sistema koja su simulirana za potrebe testiranja.

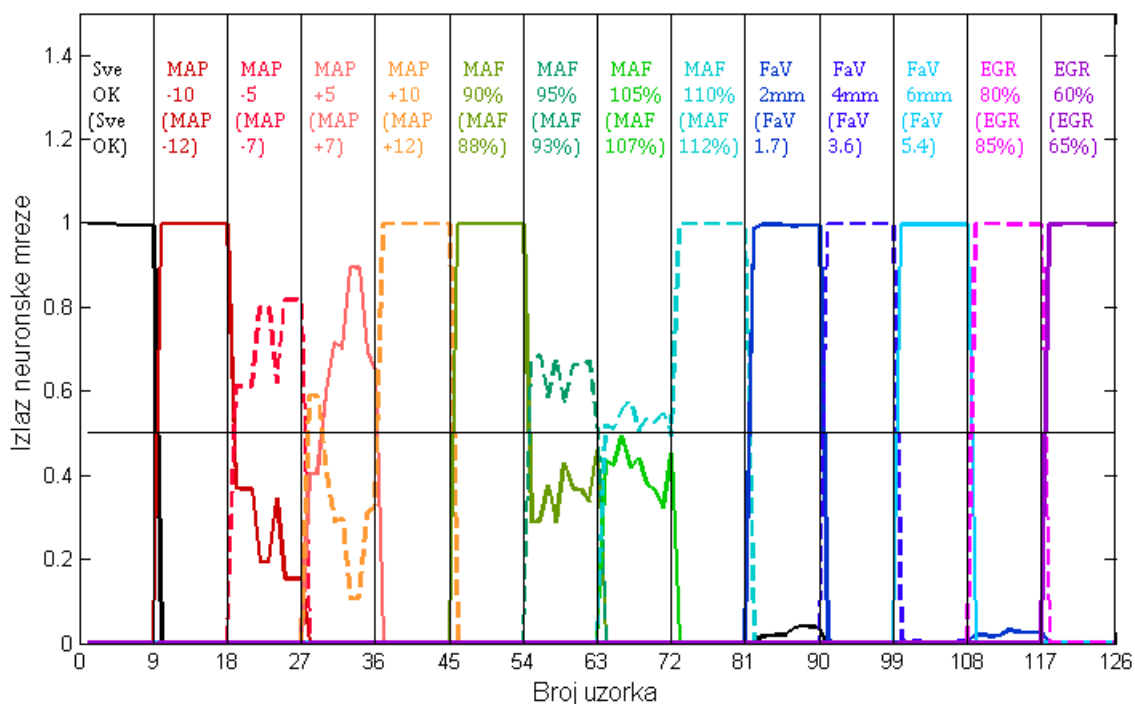
Da ne bi bilo zabune, evo konkretnog objašnjenja pomoću *Sl. 6.15b*. Ako se posmatra, na primer, deo apscise od uzorka 54 zaključno sa uzorkom 63, oznaka „(MAF93%)“ u toj oblasti dijagrama govori da su uzorci 55-63 dobijeni simuliranjem stanja „MAF93%“. Gledajući samo tu oblast na dijagramu, na osnovu linija odziva svih izlaznih neurona može se zaključiti da na neispravnost „MAF93%“ konkretna neuronska mreža reaguje preko dva svoja izlazna neurona: „MAF95%“ i „MAF90%“, što se vidi po bojama linija odziva izlaznih neurona u posmatranoj oblasti dijagrama. Linija odziva izlaznog „MAF90%“ neurona je malo ispod praga detekcije stanja, a linija odziva izlaznog „MAF95%“ neurona je iznad linije praga, dok se linije odziva svih ostalih neurona ni ne vide, jer su njihovi odzivi približno ili tačno jednaki nuli. To znači da dijagnostički model u ovom konkretnom primeru alarmira neispravnost „MAF95%“. Međutim, i da je alarmirao neispravnost „MAF90%“, moglo bi se reći da je neispravnost uspešno dijagnostikovana, jer je u dijagnostici primarni cilj identifikovanje tipa neispravnosti, a identifikovanje intenziteta neispravnosti je od sekundarnog značaja. Osim toga, ukupan broj neispravnosti istog tipa, a različitih

intenziteta, može biti ogroman, pa je u većini slučajeva praktično nemoguće prepoznati tačan intenzitet nastale neispravnosti.

Što se tiče ostalih neispravnosti, na *Sl. 6.15b* se vidi da su svi njihovi tipovi uspešno identifikovani na svim uzorcima, jer su odzivi odgovarajućih izlaznih neurona iznad vrednosti 0.5.

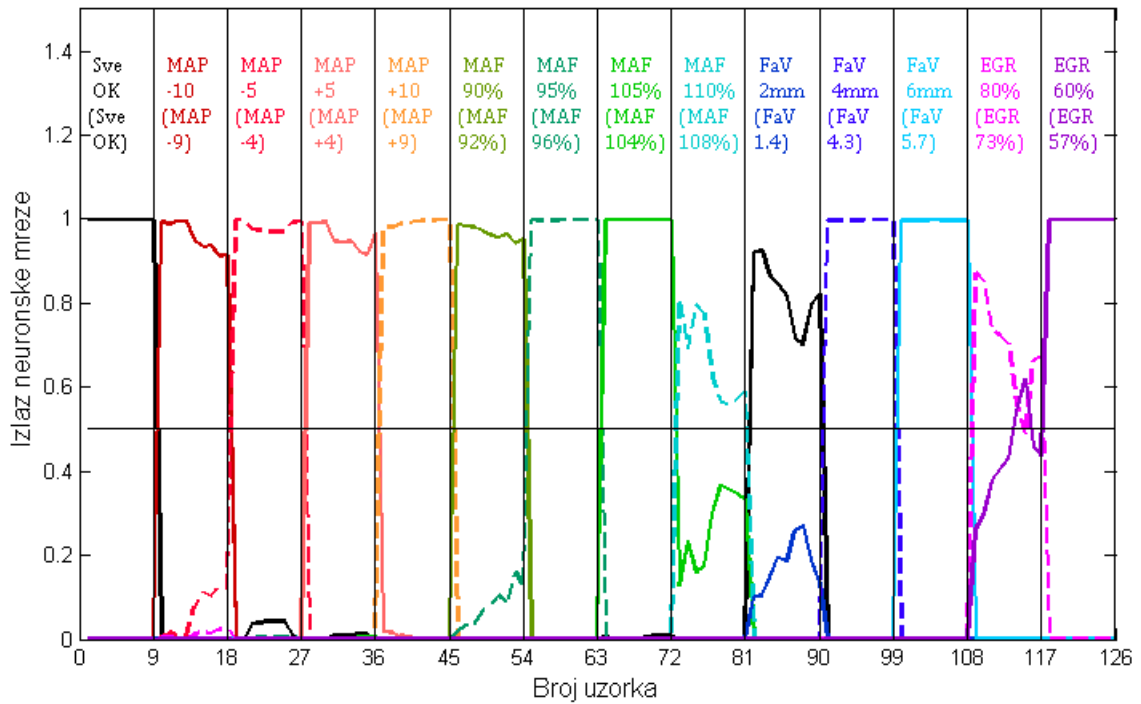


(a)

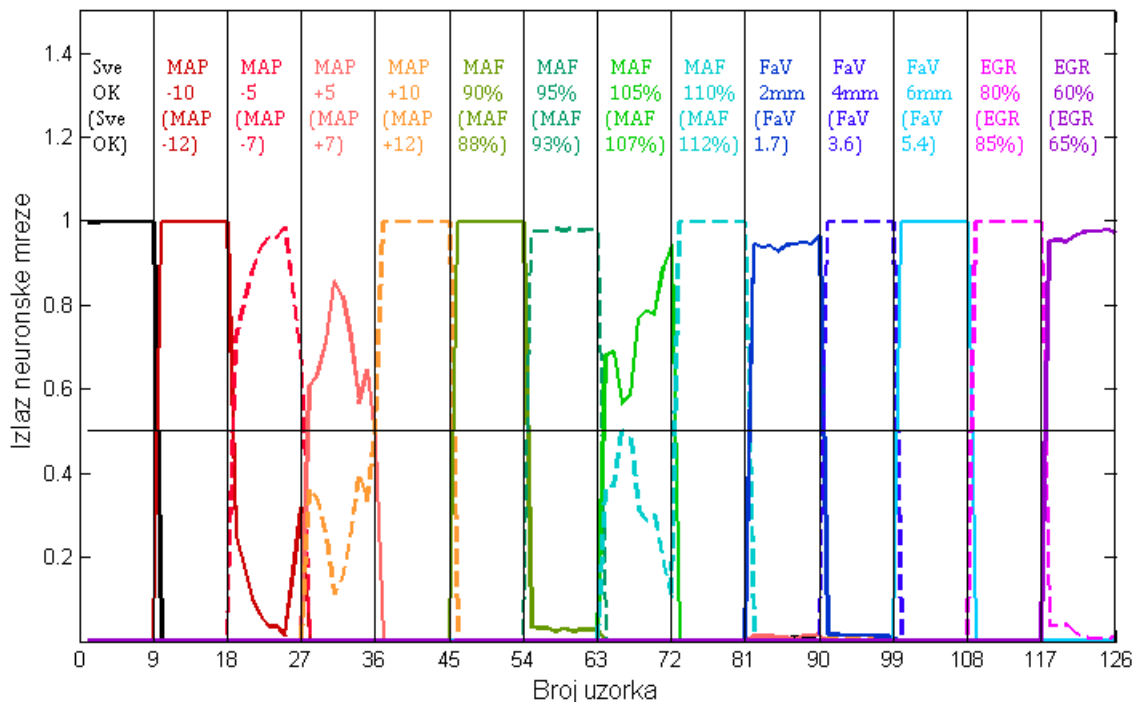


(b)

*Sl. 6.15* Rezultati testiranja neuronske mreže – motor sa masenim protokomerom vazduha, zatvorena sprega sa  $\lambda$ -sondom,  $V$  stepen prenosa, otvor leptira 17%, temperatura okoline 30°C, greška  $1.7783 \cdot 10^{-7}$ , nepoznata stanja grupe 1; (a) na bazi trenutnih vrednosti, (b) na bazi uprosečenih vrednosti



Sl. 6.16 Rezultati testiranja neuronske mreže – motor sa masenim protokomerom vazduha, zatvorena sprega sa  $\lambda$ -sondom, IV stepen prenosa, otvor leptira 13%, temperatura okoline 10°C, greška  $1.8881 \cdot 10^{-7}$ , nepoznata stanja grupe 2



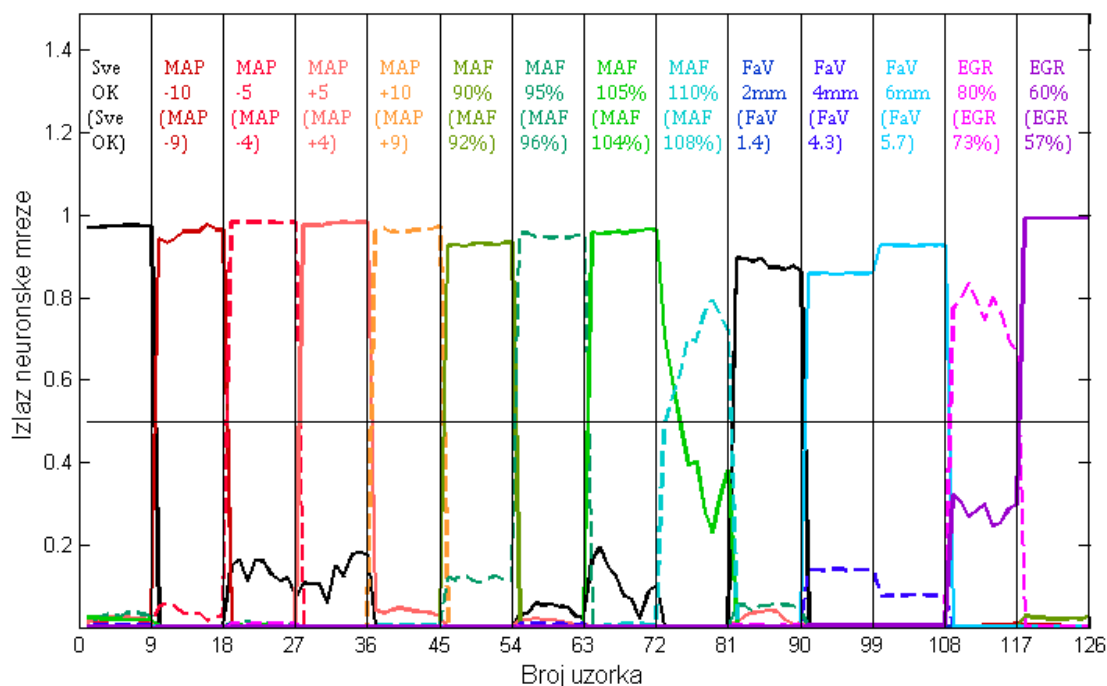
Sl. 6.17 Rezultati testiranja neuronske mreže – motor sa masenim protokomerom vazduha, otvorena sprega sa  $\lambda$ -sondom, III stepen prenosa, otvor leptira 7%, temperatura okoline 20°C, greška  $2.0731 \cdot 10^{-7}$ , nepoznata stanja grupe 1

Uz nekoliko izuzetaka, slične rezultate po pitanju uspešnosti identifikovanja stanja usisnog sistema pokazuju i Sl. 6.16 - Sl. 6.22. Konkretni izuzeci su navedeni u daljem tekstu, u komentarima koji se tiču svakog pojedinačnog dijagrama.

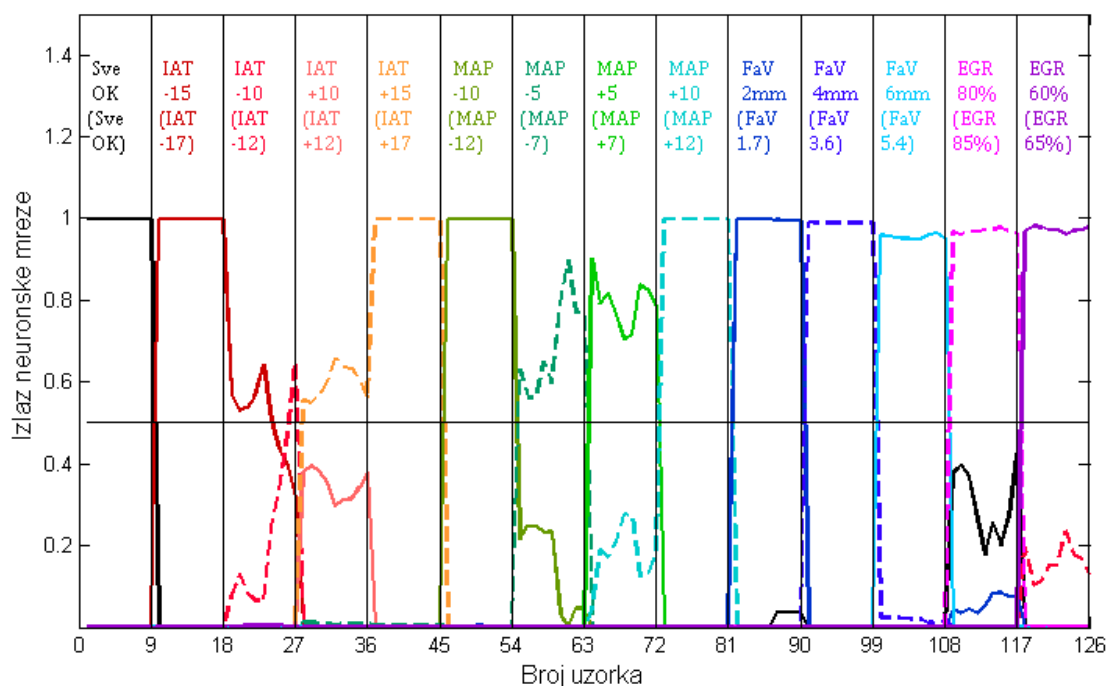
Dijagram na Sl. 6.16 pokazuje da su, osim jedne, sve neispravnosti uspešno dijagnostikovane. Taj izuzetak je neispravnost „FaV1.4“, na koju neuronska mreža

najintenzivnije reaguje svojim izlaznim „SveOK“ neuronom, signalizirajući da je sa usisnim sistemom sve u redu, iako nije tako.

U slučaju na Sl. 6.17, sve neispravnosti su uspešno dijagnostikovane.



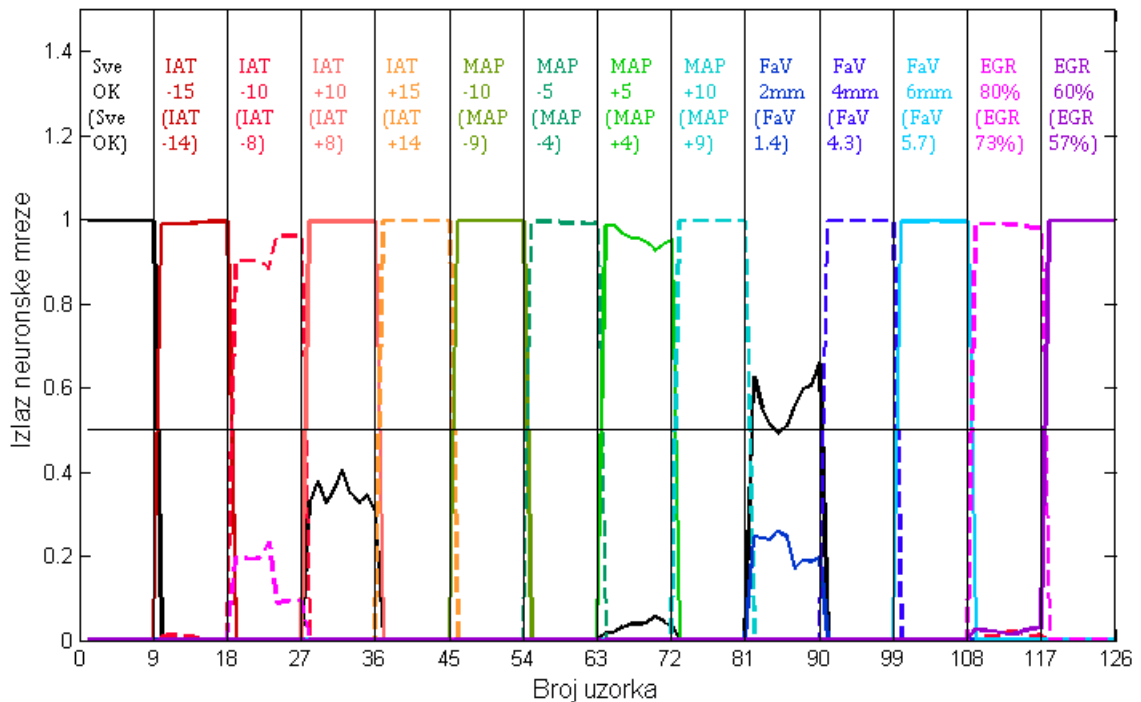
Sl. 6.18 Rezultati testiranja neuronske mreže – motor sa masenim protokomerom vazduha, otvorena sprega sa  $\lambda$ -sondom, II stepen prenosa, otvor leptira 6%, temperatura okoline 10°C, greška 0.0018526, nepoznata stanja grupe 2



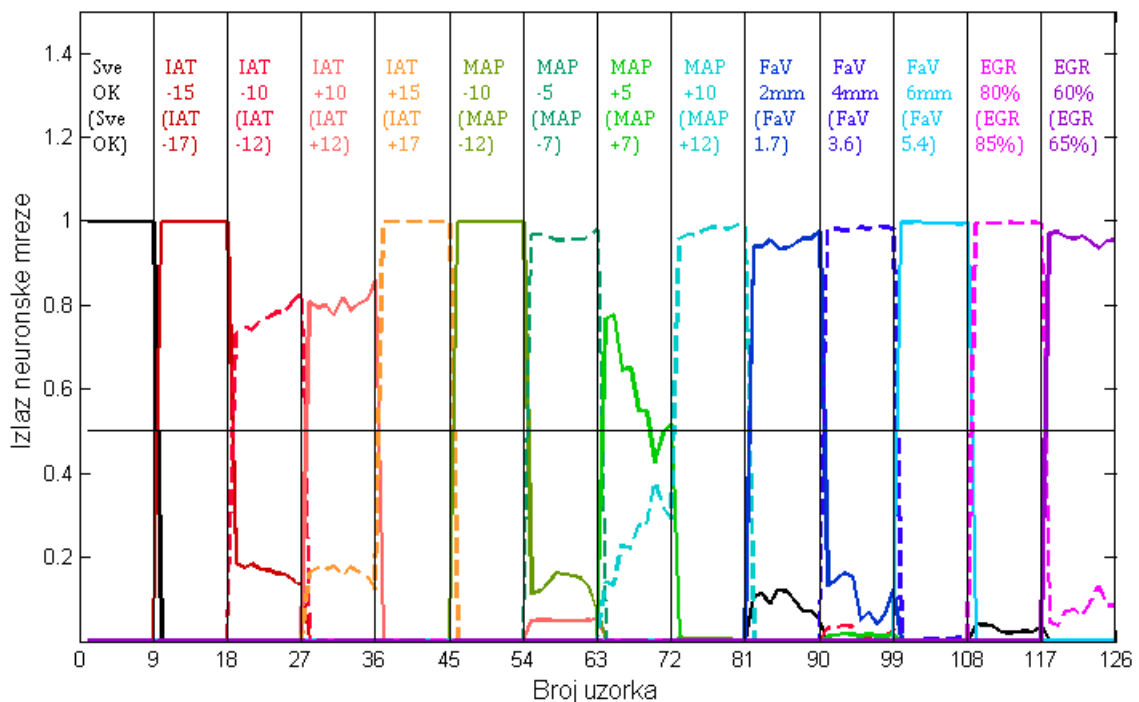
Sl. 6.19 Rezultati testiranja neuronske mreže – motor bez protokomera vazduha, zatvorena sprega sa  $\lambda$ -sondom, V stepen prenosa, otvor leptira 13%, temperatura okoline 20°C, greška  $6.8749 \cdot 10^{-7}$ , nepoznata stanja grupe 1

Na Sl. 6.18, situacija je slična onoj na Sl. 6.16 – neispravnost „FaV1.4“ nije dijagnostikovana ni na jednom uzorku, a sve ostale su uspešno dijagnostikovane na svim uzorcima.

Sl. 6.19 prikazuje uspešno dijagnostikovanje svih nepravnosti, ali ne na svim uzorcima. Naime, 25. i 26. uzorak odnose se na neispravnost „IAT-12“, ali ni jedan izlazni neuron ne reaguje na te uzorke odzivom većim od 0.5. S obzirom da u slučaju ova dva uzorka ni izlazni „SveOK“ neuron nema odziv veći od 0.5, to znači da dijagnostički model ukazuje na neku anomaliju u motoru, ali ona nije identifikovana.



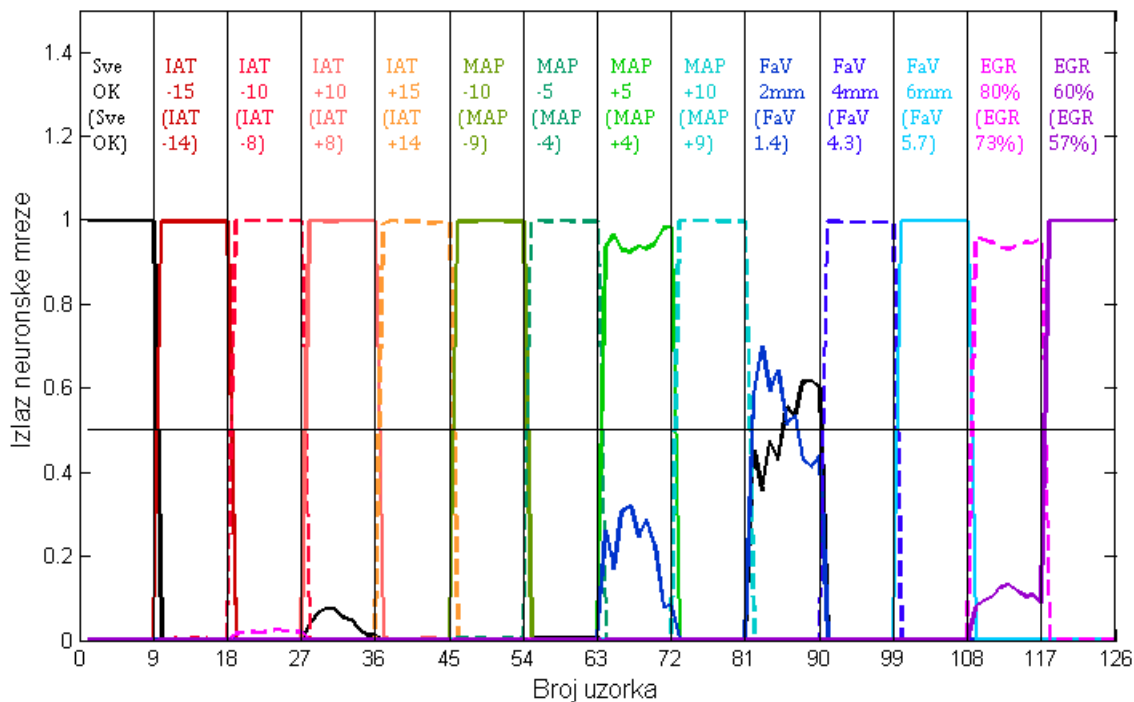
Sl. 6.20 Rezultati testiranja neuronske mreže – motor bez protokomera vazduha, zatvorena sprega sa  $\lambda$ -sondom, II stepen prenosa, otvor leptira 5%, temperatura okoline 0°C, greška  $2.4946 \cdot 10^{-7}$ , nepoznata stanja grupe 2



Sl. 6.21 Rezultati testiranja neuronske mreže – motor bez protokomera vazduha, otvorena sprega sa  $\lambda$ -sondom, III stepen prenosa, otvor leptira 8%, temperatura okoline 30°C, greška  $3.1834 \cdot 10^{-7}$ , nepoznata stanja grupe 1

Dijagram na *Sl. 6.20* prikazuje veoma uspešno dijagnostikovanje svih neispravnosti, osim „FaV1.4“. Na ovu neispravnost neuronska mreža reaguje kao da je sve u redu, sa izuzetkom 85-og uzorka. U slučaju ovog uzorka, liniju praga ne prelazi ni odziv izlaznog „SveOK“ neurona, što ukazuje da nešto nije u redu sa motorom.

Prema dijagramu na *Sl. 6.21*, samo na uzorcima 70 i 71 (neispravnost „MAP+7“) izostala je ispravna reakcija neuronske mreže, dok su neispravnosti usisnog sistema uspešno prepoznate u slučaju svih ostalih uzoraka.



*Sl. 6.22* Rezultati testiranja neuronske mreže – motor bez protokomera vazduha, otvorena sprega sa  $\lambda$ -sondom, IV stepen prenosa, otvor leptira 11%, temperatura okoline 0°C, greška  $3.1709 \cdot 10^{-7}$ , nepoznata stanja grupe 2

I na kraju, u slučaju prikazanom na *Sl. 6.22*, ponovo postoji problem sa dijagnostikovanjem neispravnosti „FaV1.4“, ali samo na nekim uzorcima (86-90), koje neuronska mreža identifikuje kao stanje bez neispravnosti. Na svim ostalim uzorcima, dijagnostikovanje neispravnosti je uspešno.

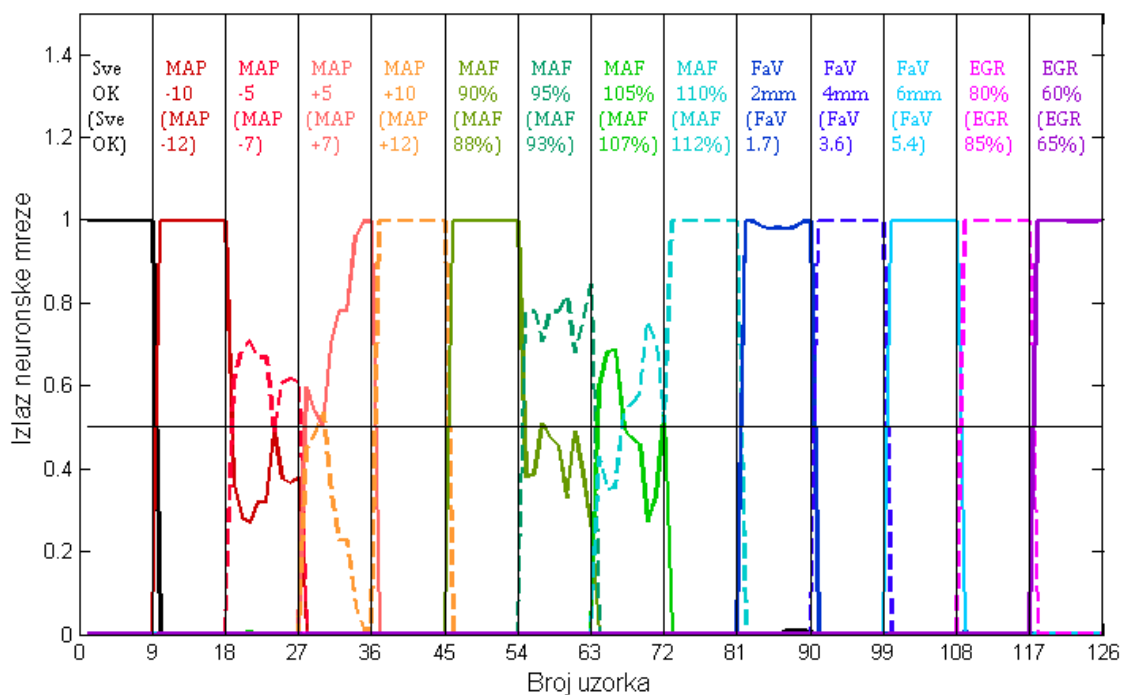
#### Nivo IV - nepoznata stanja u nepoznatim baznim radnim tačkama

Slično nivou III, i ovde su oba dijagnostička modela testirana u ukupno 64 bazne radne tačke. Zbog toga, u okviru ovog teksta neće biti priloženi svi grafički prikazi rezultata testiranja, nego se daje nekoliko reprezentativnih, koji treba da ilustruju performanse dijagnostičkih modela na ovom nivou testiranja. Pri tome je vođeno računa da budu obuhvaćena oba  $\lambda$ -režima i kod jednog i kod drugog dijagnostička modela, sa svim nepoznatim stanjima usisnog sistema i u sva četiri razmatrana stepena prenosa. Na *Sl. 6.23* - *Sl. 6.26* prikazani su dijagrami koji se odnose na slučaj usisnog sistema sa masenim protokomerom vazduha, a na *Sl. 6.27* - *Sl. 6.30* su dijagrami koji se odnose na slučaj bez protokomera. Oznake na ovim dijagramima imaju isti smisao kao i kod III nivoa testiranja.

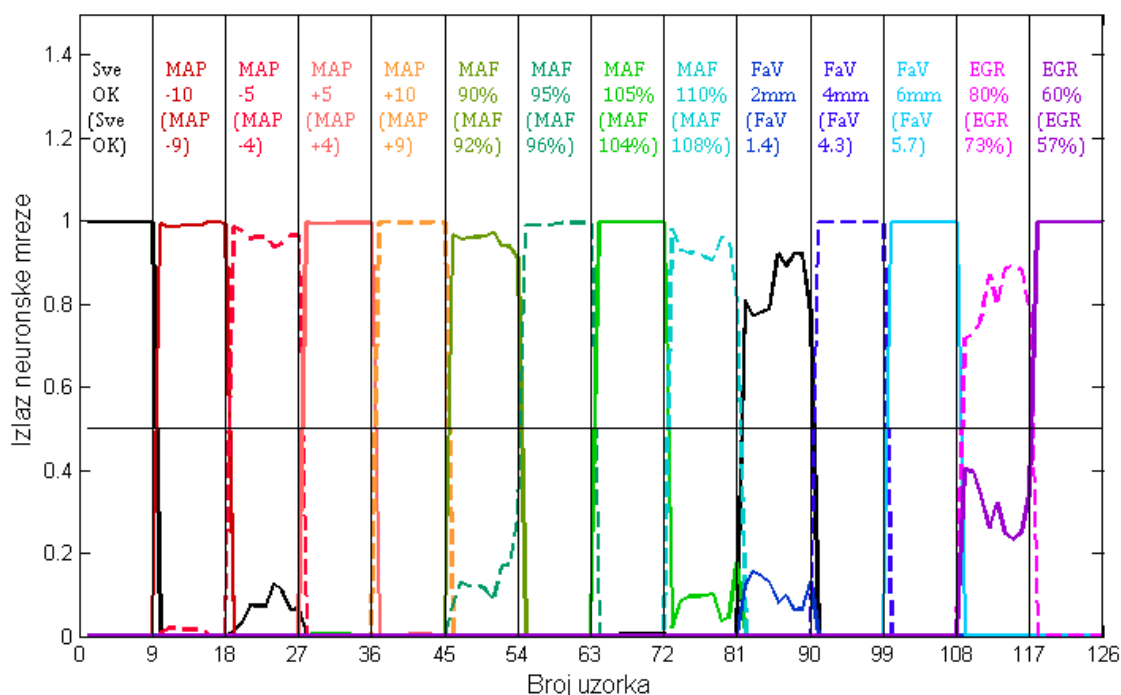
Dijagram na *Sl. 6.23* pokazuje da su sve neispravnosti uspešno dijagnostikovane, osim na 24-om uzorku, kada ni jedan izlazni neuron nema odziv iznad 0.5.

U slučaju na *Sl. 6.24*, osim neispravnosti „FaV1.4“, sve ostale su uspešno dijagnostikovane. Na slici se vidi da pri postojanju ove neispravnosti, izlazni „SveOK“ neuron ima odziv iznad linije praga za sve uzorke, pokazujući da je sa motorom sve u redu, što nije tačno.





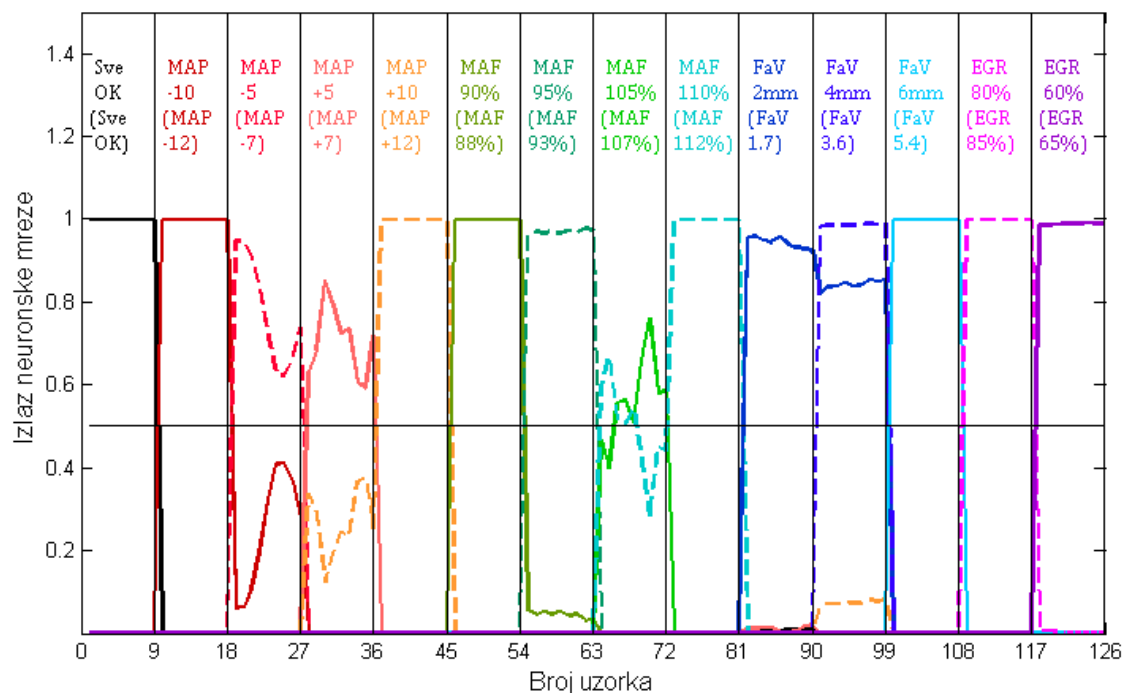
Sl. 6.23 Rezultati testiranja neuronske mreže – motor sa masenim protokomerom vazduha, zatvorena sprega sa  $\lambda$ -sondom, V stepen prenosa, otvor leptira 14%, temperatura okoline 12°C, greška  $1.7783 \cdot 10^{-7}$ , nepoznata stanja grupe 1



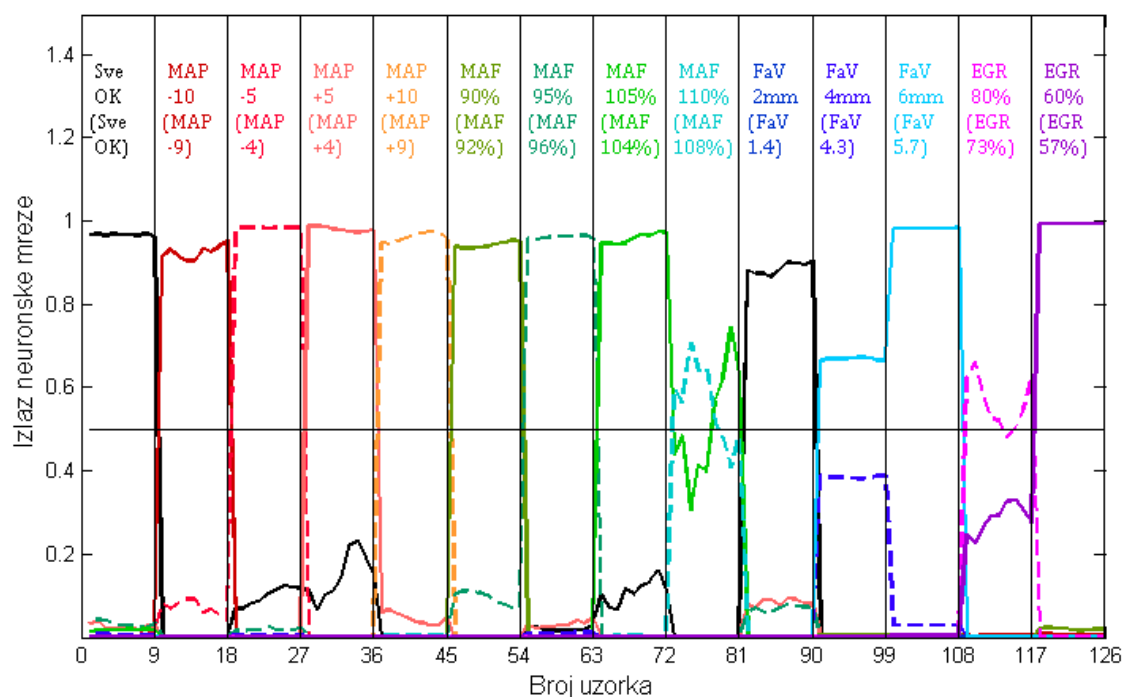
Sl. 6.24 Rezultati testiranja neuronske mreže – motor sa masenim protokomerom vazduha, zatvorena sprega sa  $\lambda$ -sondom, IV stepen prenosa, otvor leptira 12.5%, temperatura okoline 28°C, greška  $1.8881 \cdot 10^{-7}$ , nepoznata stanja grupe 2

Sl. 6.25 prikazuje uspešno dijagnostikovanje svih neispravnosti. Može se primetiti da na neke uzorke neuronska mreža reaguje tako, da dva njena izlazna neurona imaju odzive iznad 0.5. To važi za uzorke 66-68, dobijene simuliranjem neispravnosti „MAF107%“, kao i za sve uzorke dobijene simuliranjem neispravnosti „FaV3.6“. U prvom slučaju, alarm signaliziraju izlazni neuroni „MAF105%“ i „MAF110%“, a u drugom izlazni neuroni „FaV2mm“ i „FaV4mm“. Ovakvo dvostruko alarmiranje neispravnosti ne može se smatrati

slabošću dijagnostičkog modela, nego čak njegovom dobrom stranom, jer se na taj način približnije identifikuje veličina neispravnosti.



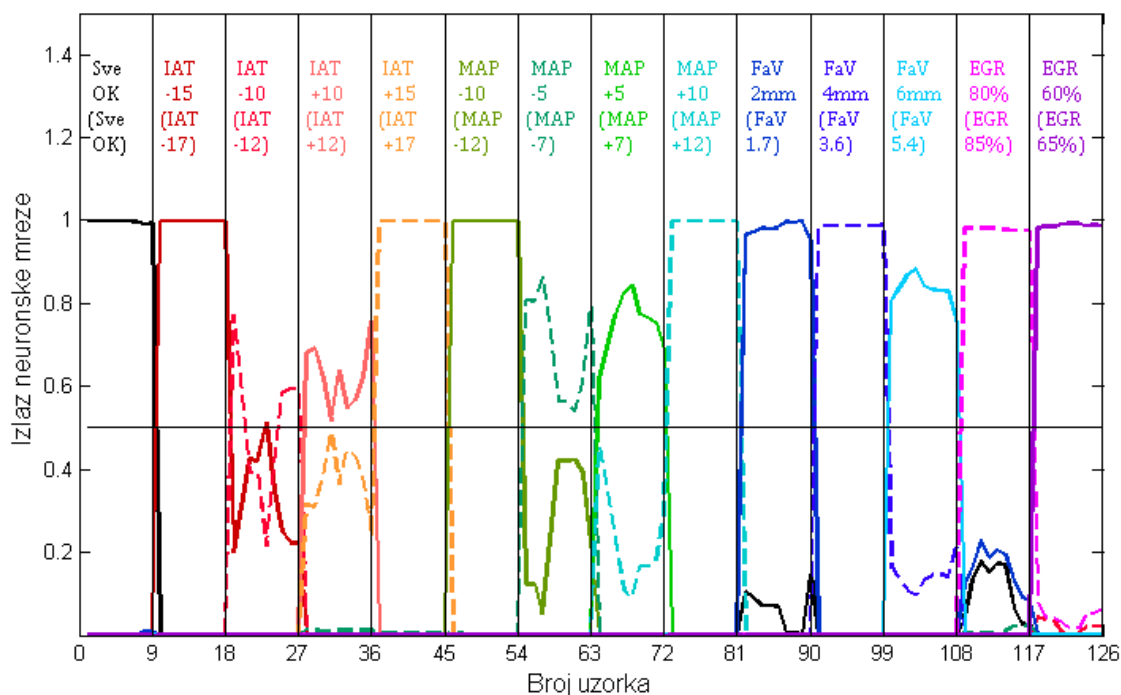
Sl. 6.25 Rezultati testiranja neuronske mreže – motor sa masenim protokomerom vazduha, otvorena sprega sa  $\lambda$ -sondom, III stepen prenosa, otvor leptira 7,4%, temperatura okoline 12°C, greška  $2.0731 \cdot 10^{-7}$ , nepoznata stanja grupe 1



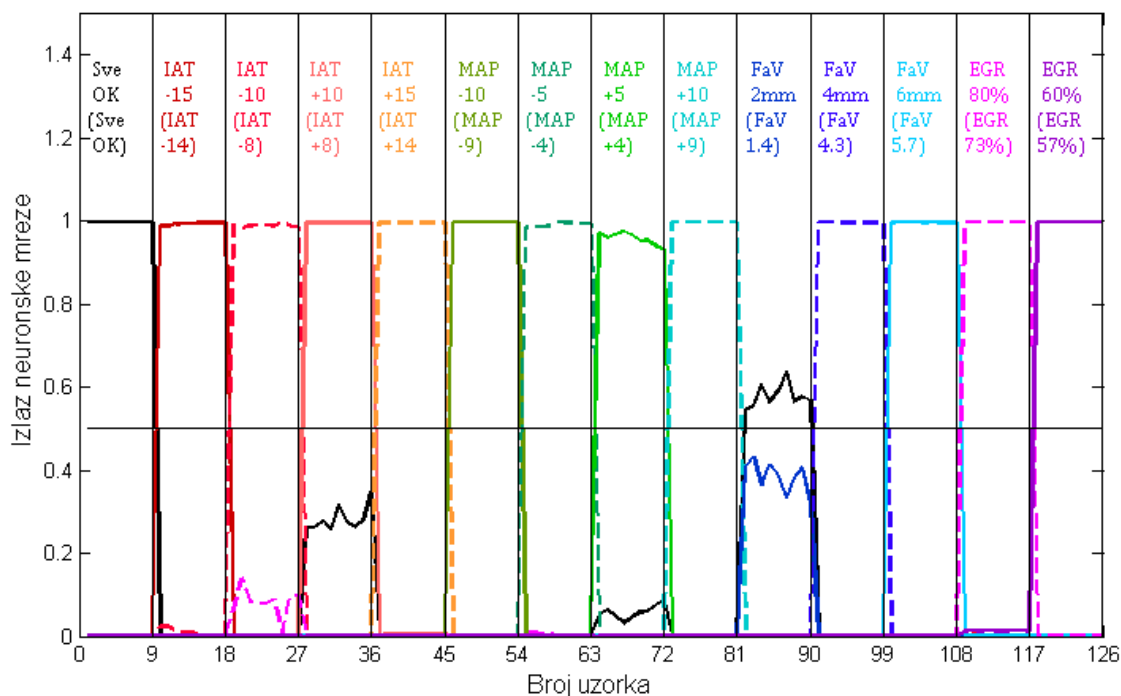
Sl. 6.26 Rezultati testiranja neuronske mreže – motor sa masenim protokomerom vazduha, otvorena sprega sa  $\lambda$ -sondom, II stepen prenosa, otvor leptira 6.7%, temperatura okoline 28°C, greška 0.0018526, nepoznata stanja grupe 2

Što se tiče slučaja na Sl. 6.26, uočavaju se neuspešna dijagnostikovanja neispravnosti „FaV1.4“ na svim uzorcima, kao i neispravnosti „EGR73%“ samo na 114-om uzorku. Osim toga, zapaža se dvostruko identifikovanje neispravnosti „MAF108%“ na uzorku broj 78, što,

kako je malo pre rečeno, doprinosi poboljšanju performansi dijagnostičkog modela. Takođe, zanimljivo je da neispravnosti „FaV4.3“ i „FaV5.7“ neuronska mreža prepoznaje kao „FaV6mm“, što se može smatrati uspešnim dijagnostikovanjem.



Sl. 6.27 Rezultati testiranja neuronske mreže – motor bez protokomera vazduha, zatvorena sprega sa  $\lambda$ -sondom, V stepen prenosa, otvor leptira 18%, temperatura okoline 28°C, greška  $6.8749 \cdot 10^{-7}$ , nepoznata stanja grupe 1

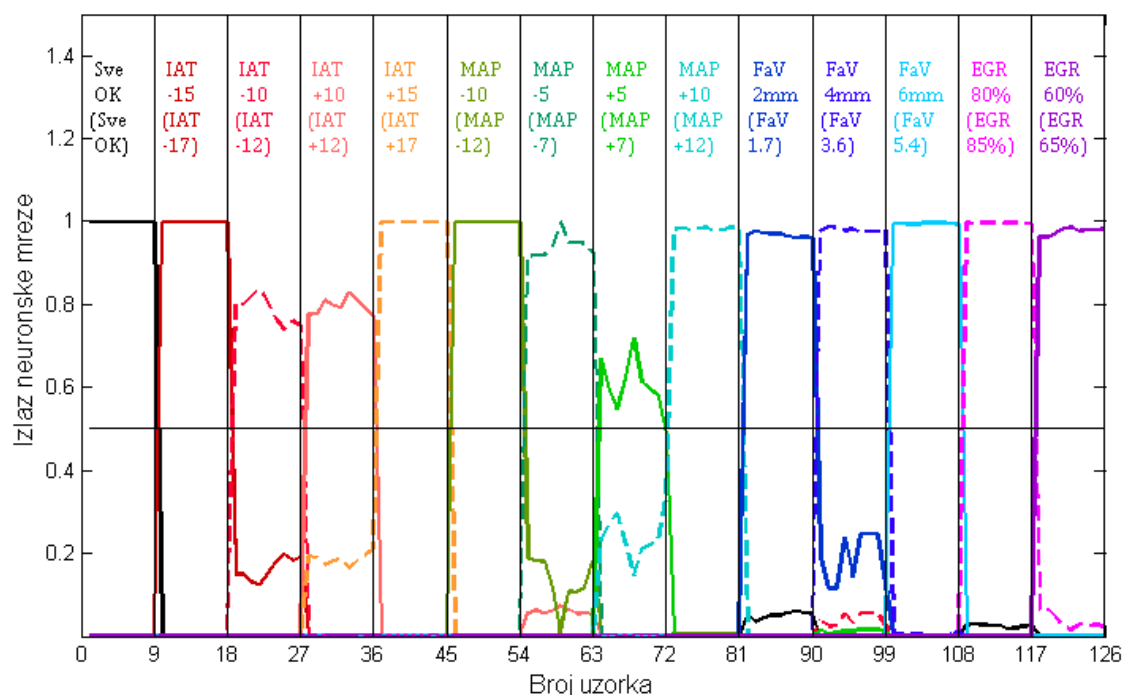


Sl. 6.28 Rezultati testiranja neuronske mreže – motor bez protokomera vazduha, zatvorena sprega sa  $\lambda$ -sondom, II stepen prenosa, otvor leptira 5.3%, temperatura okoline 12°C, greška  $2.4946 \cdot 10^{-7}$ , nepoznata stanja grupe 2

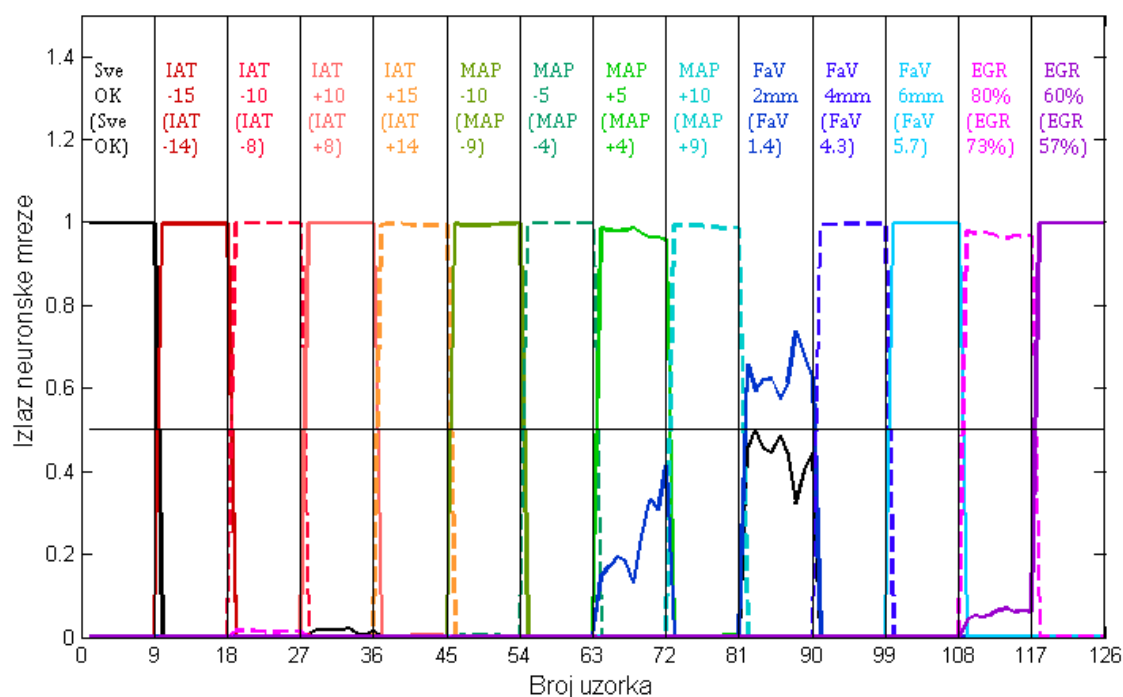
U slučaju ilustrovanom na Sl. 6.27, uspešnost dijagnostikovanja nespravnosti je potpuna, sa izuzetkom neispravnosti „IAT-12“, gde postoje tri uzorka (21, 22 i 24) bez alarma bilo

kog izlaznog neurona. S obzirom da ni izlazni „SveOK“ neuron nema odziv iznad linije praga, to znači da neuronska mreža otkriva da postoji neka neispravnost, ali je ne prepoznaje.

U slučaju na Sl. 6.28, osim neispravnosti „FaV1.4“, sve ostale su uspešno dijagnostikovane.



Sl. 6.29 Rezultati testiranja neuronske mreže – motor bez protokomera vazduha, otvorena sprega sa  $\lambda$ -sondom, III stepen prenosa, otvor leptira 8.2%, temperatura okoline 28°C, greška  $3.1834 \cdot 10^{-7}$ , nepoznata stanja grupe 1



Sl. 6.30 Rezultati testiranja neuronske mreže – motor bez protokomera vazduha, otvorena sprega sa  $\lambda$ -sondom, IV stepen prenosa, otvor leptira 10.5%, temperatura okoline 12°C, greška  $3.1709 \cdot 10^{-7}$ , nepoznata stanja grupe 2

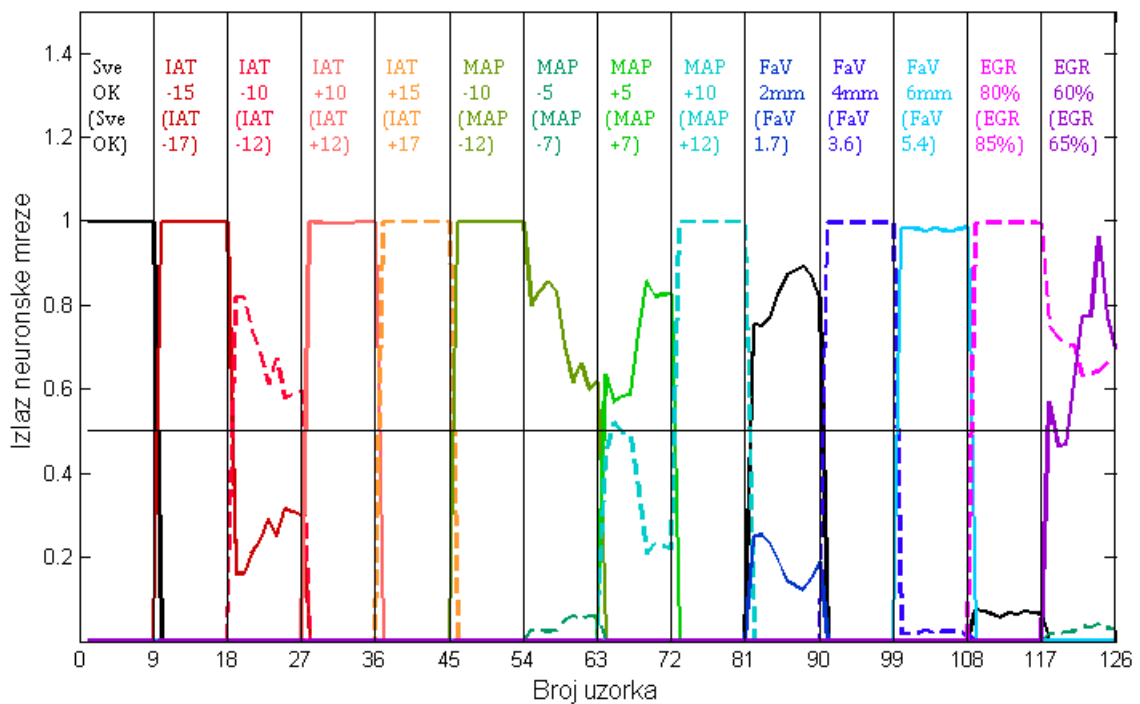
Na Sl. 6.29 i Sl. 6.30 uspešno su dijagnostikovane sve neispravnosti na svim uzorcima.

### Nivo V - nepoznata stanja u nepoznatim baznim radnim tačkama sa variranjem parametara

Broj virtuelnih eksperimenata izvedenih za potrebe testiranja na ovom, poslednjem nivou, je 32, tako da u okviru teksta ovog rada nije moguće prikazati sve dijagrame, dobijene kao rezultat testiranja. Kompletni rezultati testiranja mogu se naći na priloženom CD-ROM-u, a ovde se daje samo njihov izvod, koji bi trebalo da ilustruje u kojoj meri dijagnostički model može uspešno da obavlja svoju funkciju, ako se menjaju masa vozila, atmosferski pritisak, brzina vetra u pravcu kretanja vozila ili uzdužni nagib puta.

Na Sl. 6.31 - Sl. 6.34 prikazani su odabrani dijagrami za slučaj rada motora u zatvorenoj sprezi sa  $\lambda$ -sondom, a na Sl. 6.35 - Sl. 6.38 su dijagrami koji se odnose na rad motora u otvorenoj sprezi sa  $\lambda$ -sondom. Treba podsetiti da su svi dijagrami dobijeni na primeru motora bez protokomera vazduha u dve bazne radne tačke – jedna se odnosi na kretanje vozila u trećem, a druga na kretanje vozila u četvrtom stepenu prenosa. Izvršeno je testiranje prepoznavanja nepoznatih neispravnosti grupe 1 iz Tab. 6.7 (na strani 137).

Na Sl. 6.31 može se videti da se povećanjem mase vozila za 100 kg ne pogoršavaju u značajnoj meri performanse dijagnostičkog modela. Osim neispravnosti „FaV1.7“, sve ostale su na svim uzorcima uspešno dijagnostikovane.

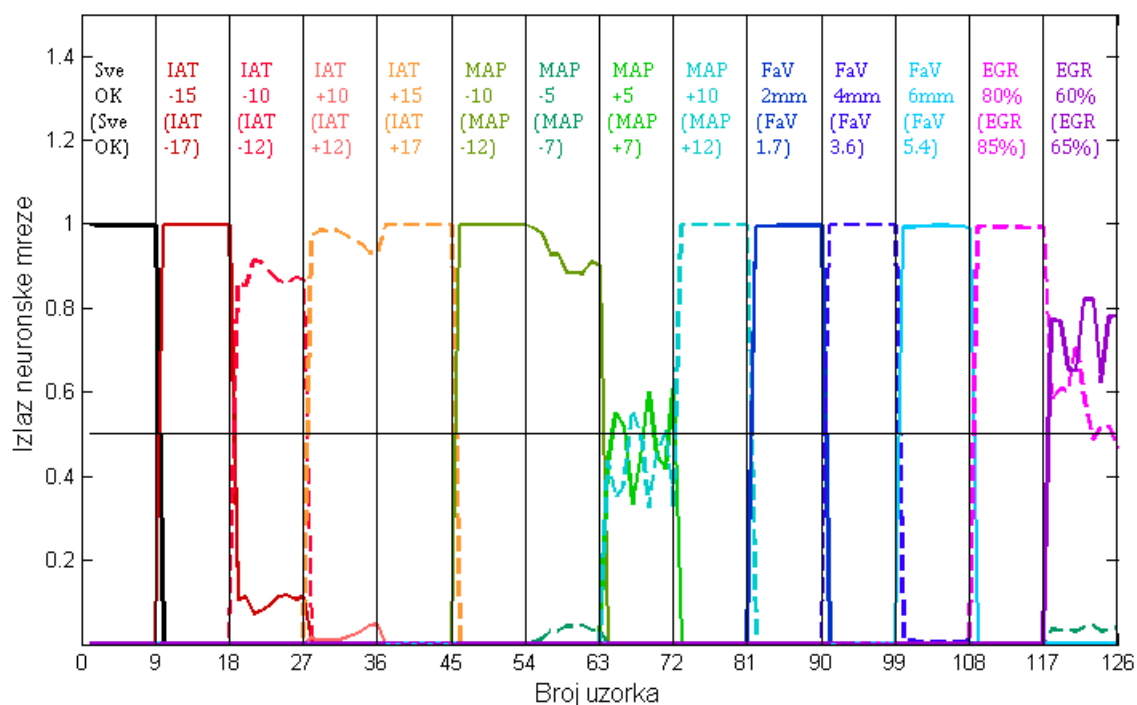


Sl. 6.31 Rezultati testiranja neuronske mreže – zatvorena spreza sa  $\lambda$ -sondom, III stepen prenosa, otvor leptira 7.4%, temperatura okoline 12°C, masa vozila 1100 kg, atmosferski pritisak 1 bar, brzina vetra 0 km/h, uzdužni nagib puta 0%

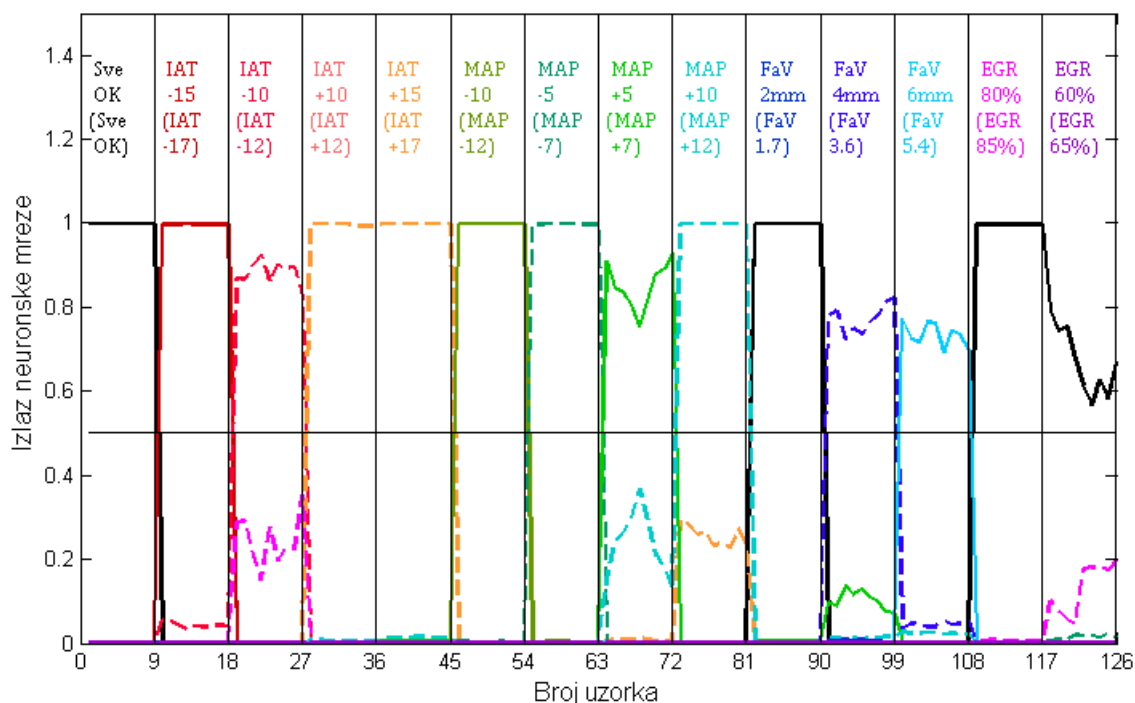
Dijagram na Sl. 6.32 ilustruje performanse dijagnostičkog modela pri povećanom atmosferskom pritisku za 2kPa u odnosu na vrednost korišćenu prilikom obuke neuronske mreže. Zapaža se uspešno dijagnostikovanje svih neispravnosti na svim uzorcima, sa izuzetkom neispravnosti „MAP+7“. Ova neispravnost je prisutna na uzorcima 64-72, a na 64-om i 70-om uzorku ni jedan izlazni neuron nema vrednost odziva veću od 0.5. Ipak, pošto ni odziv izlaznog „SveOK“ neurona nije iznad linije praga, to znači da je dijagnostički model i na ta dva uzorka detektovao da postoji neka neispravnost, ali nije prepoznao koja je u pitanju.

Uticaj vetra, koji duva brzinom 15km/h suprotno smeru kretanja vozila, na sposobnost dijagnostičkog modela da dijagnostikuje nepoznate neispravnosti, prikazan je na Sl. 6.33.

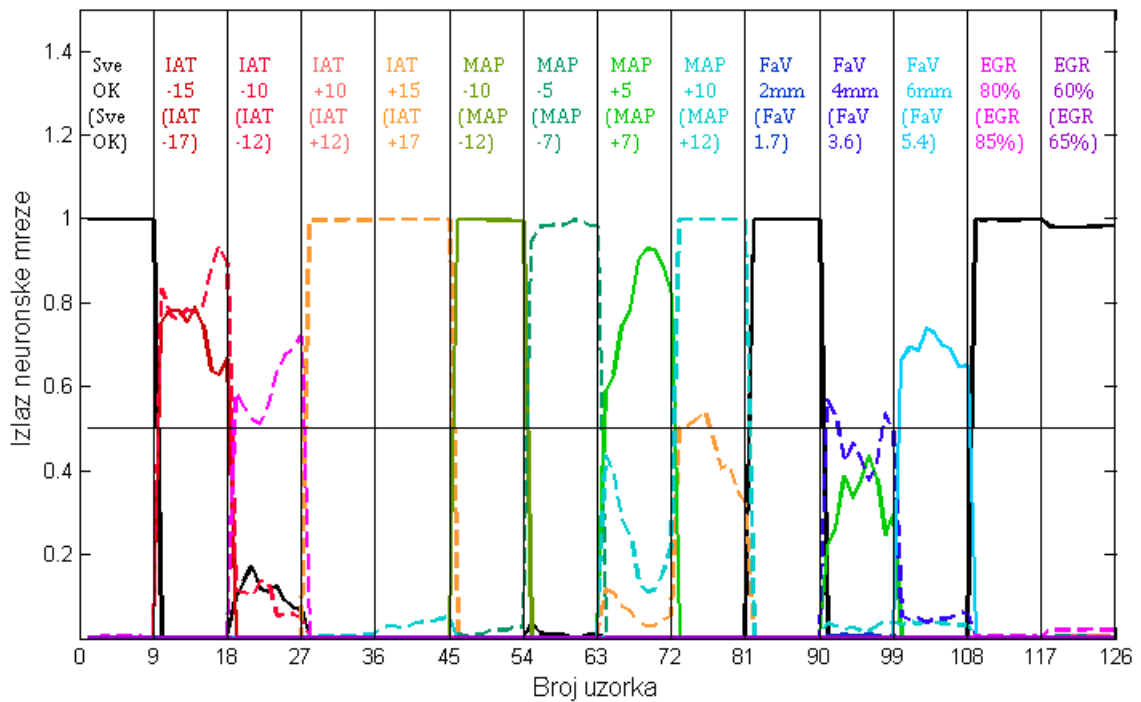
Primećuje se da neispravnosti „FaV1.7“, „EGR85%“ i „EGR65%“ dijagnostički model ne prepoznaje, jer u sva tri slučaja, izlazni „SveOK“ neuron ima vrednost veću od 0.5. Sve ostale neispravnosti su uspešno dijagnostikovane na svim uzorcima.



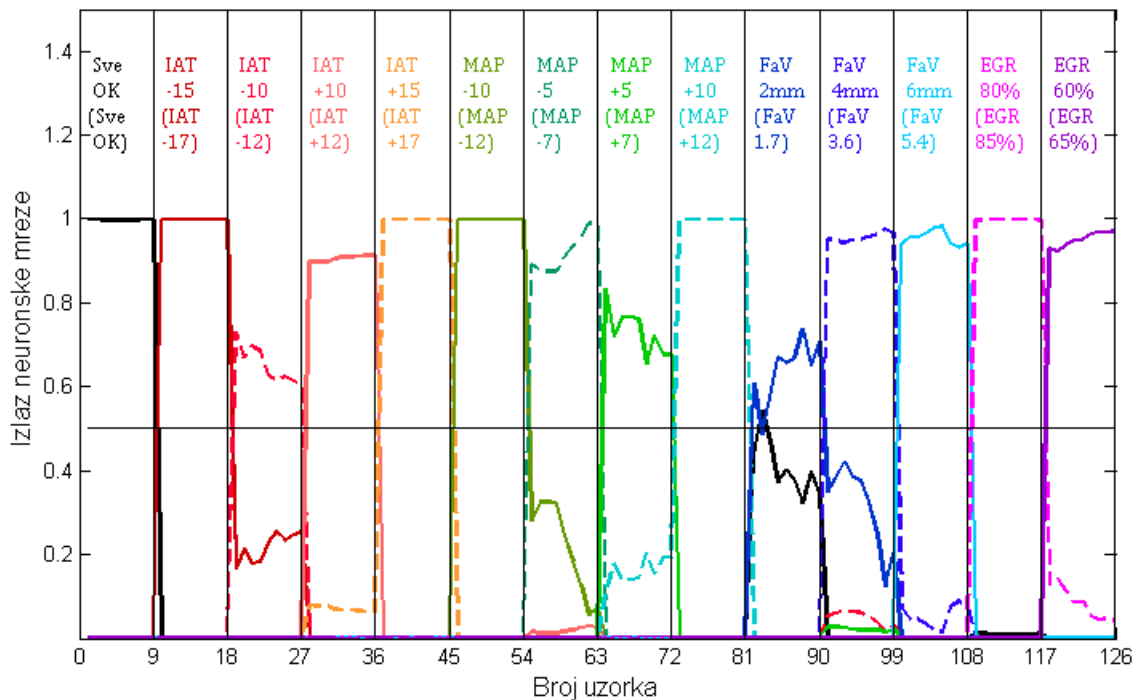
Sl. 6.32 Rezultati testiranja neuronske mreže – zatvorena sprega sa  $\lambda$ -sondom, III stepen prenosa, otvor leptira 7.4%, temperatura okoline 12°C, masa vozila 1000 kg, atmosferski pritisak 1.02 bar, brzina vetra 0 km/h, uzdužni nagib puta 0%



Sl. 6.33 Rezultati testiranja neuronske mreže – zatvorena sprega sa  $\lambda$ -sondom, IV stepen prenosa, otvor leptira 10.5%, temperatura okoline 12°C, masa vozila 1000 kg, atmosferski pritisak 1 bar, brzina vetra +15 km/h, uzdužni nagib puta 0%



Sl. 6.34 Rezultati testiranja neuronske mreže – zatvorena sprega sa  $\lambda$ -sondom, IV stepen prenosa, otvor leptira 10.5%, temperatura okoline 12°C, masa vozila 1000 kg, atmosferski pritisak 1 bar, brzina vetra 0 km/h, uzdužni nagib puta +1%

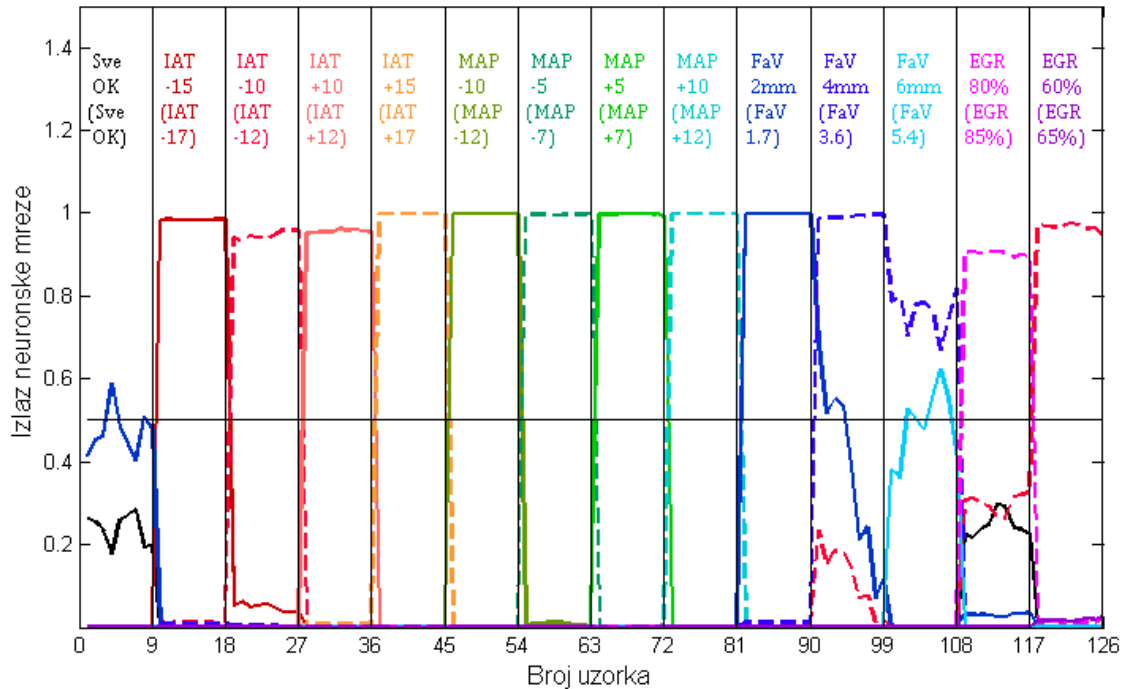


Sl. 6.35 Rezultati testiranja neuronske mreže – otvorena sprega sa  $\lambda$ -sondom, III stepen prenosa, otvor leptira 7.4%, temperatura okoline 12°C, masa vozila 1000 kg, atmosferski pritisak 0.98 bar, brzina vetra 0 km/h, uzdužni nagib puta 0%

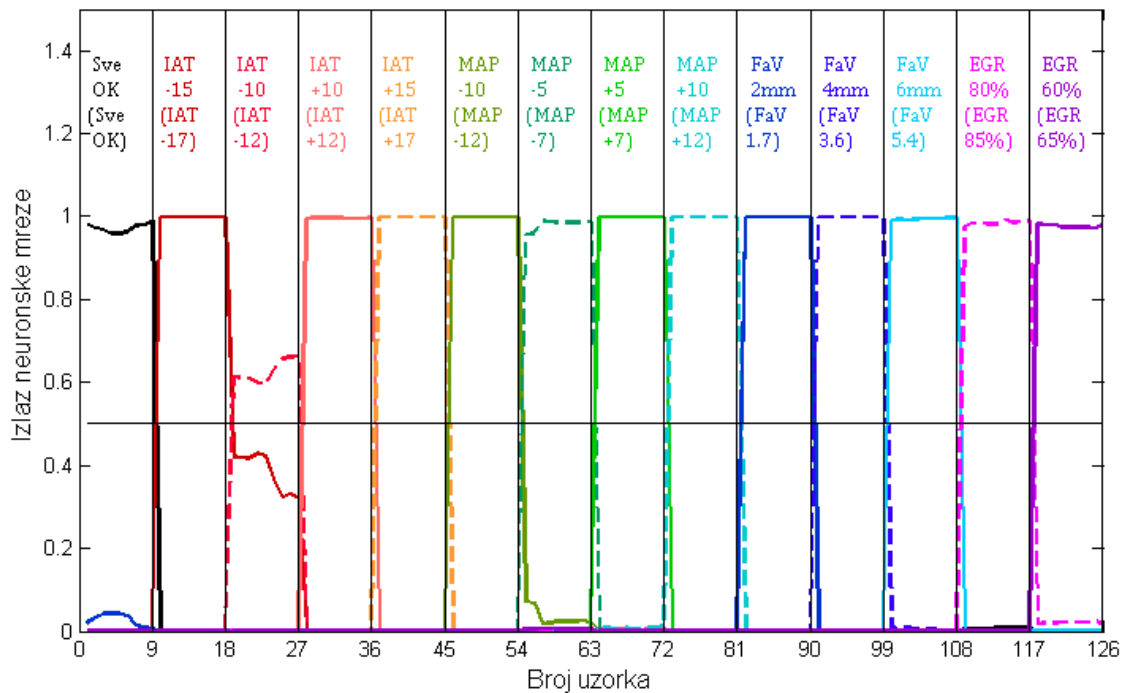
Sl. 6.34 pokazuje da povećanje uzdužnog nagiba puta za 1% značajno pogoršava performanse dijagnostičkog modela. Neispravnosti „FaV1.7“, „EGR85%“ i „EGR65%“ uopšte nisu registrovane kao neispravnosti. Dalje, pojavu neispravnosti „IAT-12“ model vidi kao „EGR85%“ na svim uzorcima, a neispravnost „FaV3.6“ nije prepoznata na uzorcima 93-97 i 99. Ostale neispravnosti su prepoznate na svim uzorcima, s tim što se u slučaju

neispravnosti „MAP+12“ pojavljuje i lažni alarm izlaznog „IAT+17“ neurona na uzorcima 73-76.

Dijagram na Sl. 6.35 prikazuje uspešno dijagnostikovanje svih neispravnosti na svim uzorcima, osim na jednom (uzorak 83). Ovaj dijagram odnosi se na slučaj kada je atmosferski pritisak za 2kPa niži od onog, koji je korišćen pri dobijanju podataka za obuku neuronskih mreža.



Sl. 6.36 Rezultati testiranja neuronske mreže – otvorena sprega sa  $\lambda$ -sondom, III stepen prenosa, otvor leptira 7.4%, temperatura okoline 12°C, masa vozila 1000 kg, atmosferski pritisak 1 bar, brzina vetra –15 km/h, uzdužni nagib puta 0%



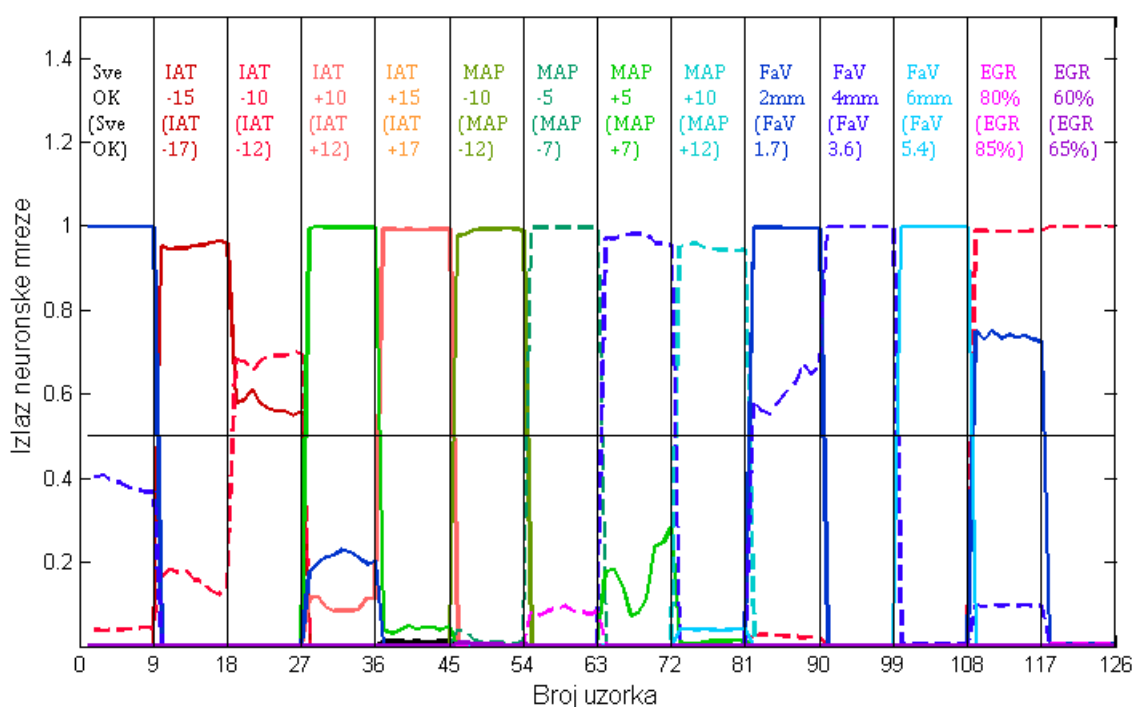
Sl. 6.37 Rezultati testiranja neuronske mreže – otvorena sprega sa  $\lambda$ -sondom, IV stepen prenosa, otvor leptira 7.4%, temperatura okoline 12°C, masa vozila 900 kg, atmosferski pritisak 1 bar, brzina vetra 0 km/h, uzdužni nagib puta 0%



Na Sl. 6.36 prikazano je kako vetar, koji duva brzinom od 15km/h u smeru kretanja vozila, utiče na performanse dijagnostičkog modela. Sve neispravnosti, osim jedne, dijagnostikovane su na svim uzorcima. Izuzetak je neispravnost „EGR65%“, koju model nije prepoznao ni na jednom uzorku odnosno ona je pogrešno prepoznata kao „IAT-12“. Problem postoji i pri identifikaciji stanja „SveOK“, koje nije identifikovano izlaznim „SveOK“ neuronom, a na uzorcima 4 i 8 pojavljuje se lažni „FaV2mm“ alarm.

Umanjenje mase vozila za 100kg u odnosu na vrednost korišćenu pri obuci neuronskih mreža, ne pogoršava performanse dijagnostičkog modela, što je ilustrovano na Sl. 6.37. Na slici se jasno vidi da su sve neispravnosti uspešno dijagnostikovane na svim uzorcima.

I na kraju, Sl. 6.38 pokazuje da bi vožnja niz nagib od 1%, značajno pogoršala karakteristike dijagnostičkog modela koji je razvijen uz pretpostavku da se vozilo kreće po horizontalnom putu. Pogrešno su dijagnostikovane neispravnosti: „IAT+12“, „MAP+7“, „EGR85%“ i „EGR65%“, kao i stanje „SveOK“. Sve ostale neispravnosti su tačno dijagnostikovane na svim uzorcima.



Sl. 6.38 Rezultati testiranja neuronske mreže – otvorena sprega sa  $\lambda$ -sondom, IV stepen prenosa, otvor leptira 7.4%, temperatura okoline 12°C, masa vozila 1000 kg, atmosferski pritisak 1 bar, brzina vetra 0 km/h, uzdužni nagib puta -1%

### 6.3.8 Zaključni komentar rezultata testiranja

Testiranje razvijenog dijagnostičkog modela, zasnovanog na podacima dobijenim iz virtuelnih eksperimenata, dalo je veoma dobre rezultate i za motor sa direktnim merenjem masenog protoka vazduha i za motor sa izračunavanjem tog protoka. Zaključci, koji se daju u nastavku odnose se i na motor sa masenim protokomerom i na motor bez protokomera.

Pokazalo se da dijagnostički model bez greške prepoznaje sve neispravnosti, koje su istog tipa i veličine, kao one koje su korišćene prilikom obuke dijagnostičkog modela. Ovo važi, kako za one režime rada motora, koji su korišćeni prilikom obuke dijagnostičkog modela, tako i za nepoznate režime rada sa kojima se dijagnostički model prvi put susreo tokom testiranja. Naravno, u ovom drugom slučaju radi se o režimima rada motora koji su

ili unutar dijapazona korišćenog tokom obuke ili, ako su izvan tog dijapazona, onda nisu mnogo udaljeni od njegovih granica.

Kada je u pitanju testiranje dijagnostičkog modela neispravnostima koje se po veličini (intenzitetu) razlikuju od onih, koje su korišćene prilikom obuke modela, takođe su dobijeni odlični rezultati i ta konstatacija se odnosi i na poznate i na nepoznate režime rada motora. To znači da su na gotovo svim uzorcima za testiranje (preko 99%) prepoznati svi tipovi neispravnosti, sa izuzetkom ulaska „falš“-vazduha u usisni sistem kroz veoma mali otvor (prečnika 1.4 mm), koji dijagnostički sistem uglavnom nije detektovao. U slučaju ostalih nepoznatih neispravnosti, ne samo da su prepoznati svi tipovi, nego su i njihove veličine vrlo približno dijagnostikovane.

Na kraju, pri rešavanju najtežeg dijagnostičkog zadatka, gde je dijagnostički model trebalo da dijagnostikuje neispravnosti nepoznatih veličina, u slučajevima kada neki parametri okoline i parametri kretanja automobila imaju drugačije vrednosti od onih, koje su korišćene za obuku modela, rezultati su bili zadovoljavajući u većini slučajeva. Pokazalo se da promena mase vozila u nekim realnim granicama ( $\pm 100 \text{ kg}$ ), kao i varijacije pritiska okoline u realnim granicama ( $\pm 2 \text{ kPa}$ ) ne pogoršavaju performanse razvijenog dijagnostičkog modela u nekoj značajnoj meri. Brzina vetra od ( $\pm 15 \text{ m/s}$ ) utiče na sposobnost dijagnostičkog modela da prepozna pojedine tipove neispravnosti (umanjeni EGR protok i ulaz „falš“-vazduha kroz mali otvor), ali su sve druge neispravnosti uspešno dijagnostikovane u slučaju pomenutog intenziteta vetra. Uzdužni nagib ima najveći uticaj na performanse dijagnostikog modela. Čak i relativno mala njegova vrednost ( $\pm 1\%$ ) dovodi do toga da dijagnostički model ne može da prepozna 5 odnosno 6 od mogućih 14 neispravnosti ni za jedan uzorak iz skupa za testiranje. Preostalih 8 odnosno 9 neispravnosti dijagnostički model uspešno prepoznaje, ali se to ne može nazvati zadovoljavajućim rezultatom. Stoga sledi zaključak da bi uzdužni nagib vozila trebalo da bude jedna od ulaznih veličina za neuronske mreže iz dijagnostičkog modela. U stvari, u Odeljku 6.3.1 ovaj parametar je i bio svrstan u ulazne veličine, ali je morao biti uklonjen iz razloga koji su opisani u tom odeljku.

## 7 DIJAGNOSTIČKI MODEL ZASNOVAN NA LABORATORIJSKI DOBIJENIM PODACIMA

### 7.1 Uvodne napomene

Dobri rezultati dijagnostičkog modela prikazanog u Poglavlju 6, na svim nivoima testiranja, predstavljaju na neki način potvrdu ispravnosti dijagnostičkog koncepta, na kojem je taj model razvijen. Iako je dijagnostički model baziran na podacima dobijenim pomoću virtuelnih eksperimanata, uspešnost dijagnostikovanja nepoznatih neispravnosti u nepoznatim baznim radnim tačkama daje za pravo da se takvi rezultati mogu smatrati nekim vidom njegove verifikacije. Međutim, bez obzira na to, pojavila se potreba za razvojem dijagnostičkog modela, zasnovanog na podacima dobijenim merenjima na stvarnom motoru, kako bi predloženi dijagnostički koncept dobio još čvršće uporište. Zbog jednostavnijeg izražavanja u daljem tekstu, na ovom mestu će biti uvedeni kompaktniji nazivi za pomenuta dva dijagnostička modela: *simulacioni dijagnostički model* (za model zasnovan na virtuelnim eksperimentima) i *laboratorijski dijagnostički model* (za model zasnovan na stvarnim merenjima). Iz tehničkih razloga, laboratorijski dijagnostički model, razvijen tokom istraživanja, nije mogao biti potpuno jednak simulacionom dijagnostičkom modelu iz Poglavlja 6, ali mu je veoma sličan, s obzirom da se zasnivaju na istom konceptu. U tom smislu, eventualni pozitivni rezultati testiranja laboratorijskog dijagnostičkog modela, mogli bi da doprinesu kompletiranju procesa verifikacije predloženog dijagnostičkog koncepta.

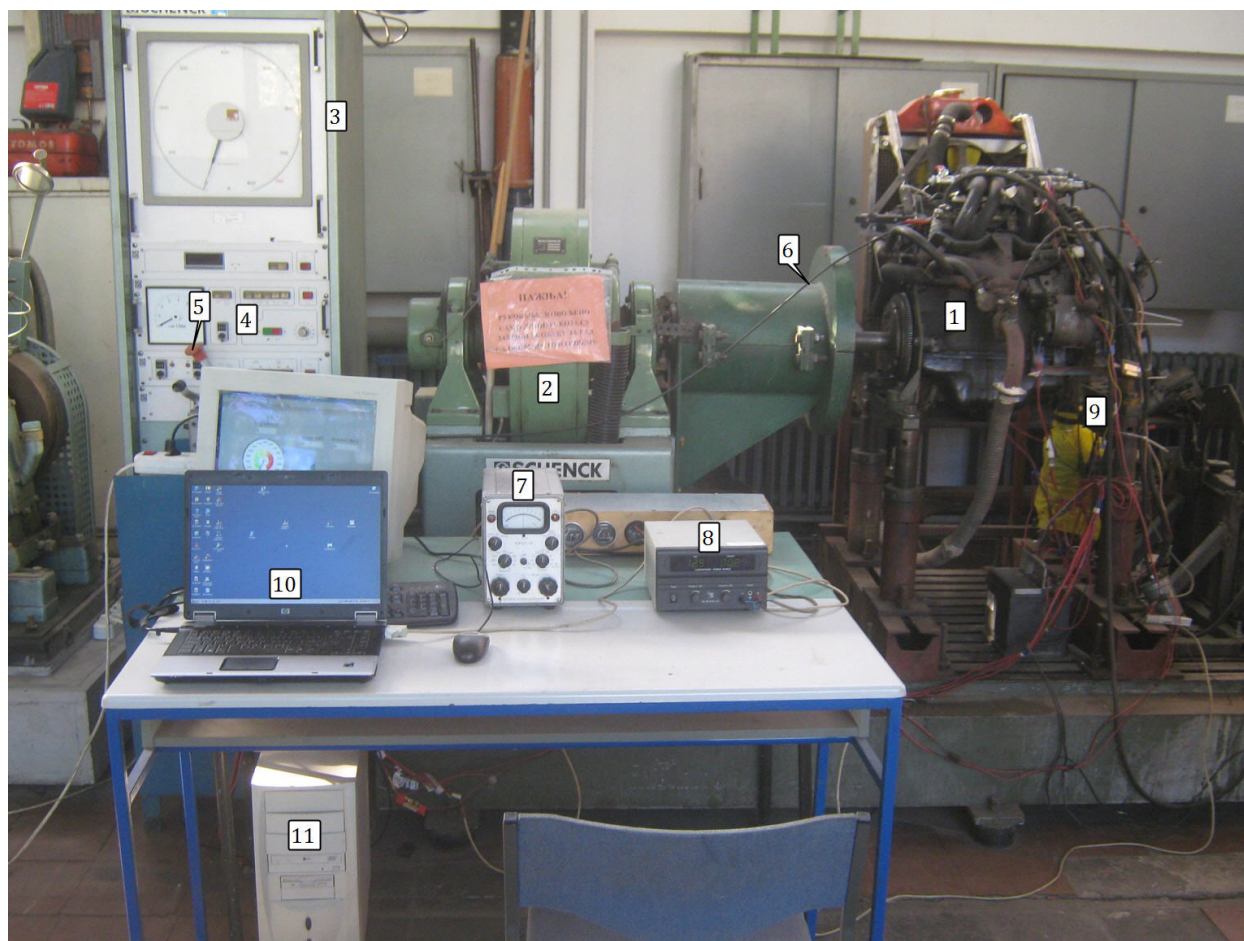
Razlike između laboratorijskog i simulacionog dijagnostičkog modela, pomenute u prethodnom pasusu, odnose se uglavnom na pokrivenost radnog opsega motora, kao i na broj posmatranih stanja usisnog sistema. Razumljivo je da će laboratorijski dijagnostički model obuhvatiti znatno uži deo radnog opsega motora u odnosu na simulacioni dijagnostički model. Razlog je znatno komplikovanija fizička realizacija realnog eksperimenta u poređenju sa virtuelnim eksperimentom, što naročito dolazi do izražaja kada treba varirati vrednosti nekih parametara. Još kada se uzme u obzir da na realnom motoru treba izazivati neispravnosti u njegovom usisnom sistemu, pa varirati intenzitete tih neispravnosti, postaje više nego jasno zašto su broj radnih tačaka i broj posmatranih neispravnosti u realnim eksperimentima značajno manji nego kod virtuelnih eksperimenata. Na kraju krajeva, izazivanje neispravnosti na realnom motoru, naročito ako se ponavlja više puta, može da ima pogubne posledice po sâm motor, dok izazivanje neispravnosti pomoću ranije opisanog simulatora neispravnosti nema nikakve štetne posledice, bez obzira na intenzitet tih neispravnosti i učestanost njihovog ponavljanja. Ova konstatacija daje još veći značaj simulatoru neispravnosti koji je razvijen u ovom radu.

Zbog navedenih poteškoća u vezi sa izvođenjem specifičnih eksperimenata na realnom motoru, laboratorijski dijagnostički model će sadržati samo jednu neuronsku mrežu, koja se odnosi na samo jedan stepen prenosa i na rad motora u zatvorenoj sprezi sa lambda-sondom, za razliku od simulacionog dijagnostičkog modela, koji se sastojao od osam neuronskih mreža. Osim toga, broj neispravnosti koje će se posmatrati, biće dvostruko manji u odnosu na simulacioni dijagnostički model, što će uticati na smanjenje broja izlaznih veličina, tj broja izlaznih neurona neuronske mreže. Ovo su glavne razlike pomenuta dva dijagnostička modela, a postoje i još neke manje razlike, koje će biti pomenute kada bude prikazivan konkretan laboratorijski dijagnostički model.

Eksperimentalna merenja u cilju prikupljanja podataka za obuku i testiranje neuronske mreže laboratorijskog dijagnostičkog modela, kao i sva neophodna pripremna merenja, izvršena su u Laboratoriji za motore i vozila Fakulteta tehničkih nauka u Novom Sadu. Opis eksperimentalnog postrojenja, sa uređajima i opremom koji su korišćeni u ove svrhe, sledi u nastavku.

## 7.2 Opis eksperimentalnog postrojenja u celini

Eksperimentalno postrojenje, koje je korišćeno u ovom istraživanju, u velikoj meri se zasniva na eksperimentalnom postrojenju koje je predstavljeno u referenci [116]. Na tom postrojenju izvršene su samo određene sitnije modifikacije, za potrebe specifičnog istraživanja u ovom radu, a njegov konačan izgled prikazan je na Sl. 7.1.



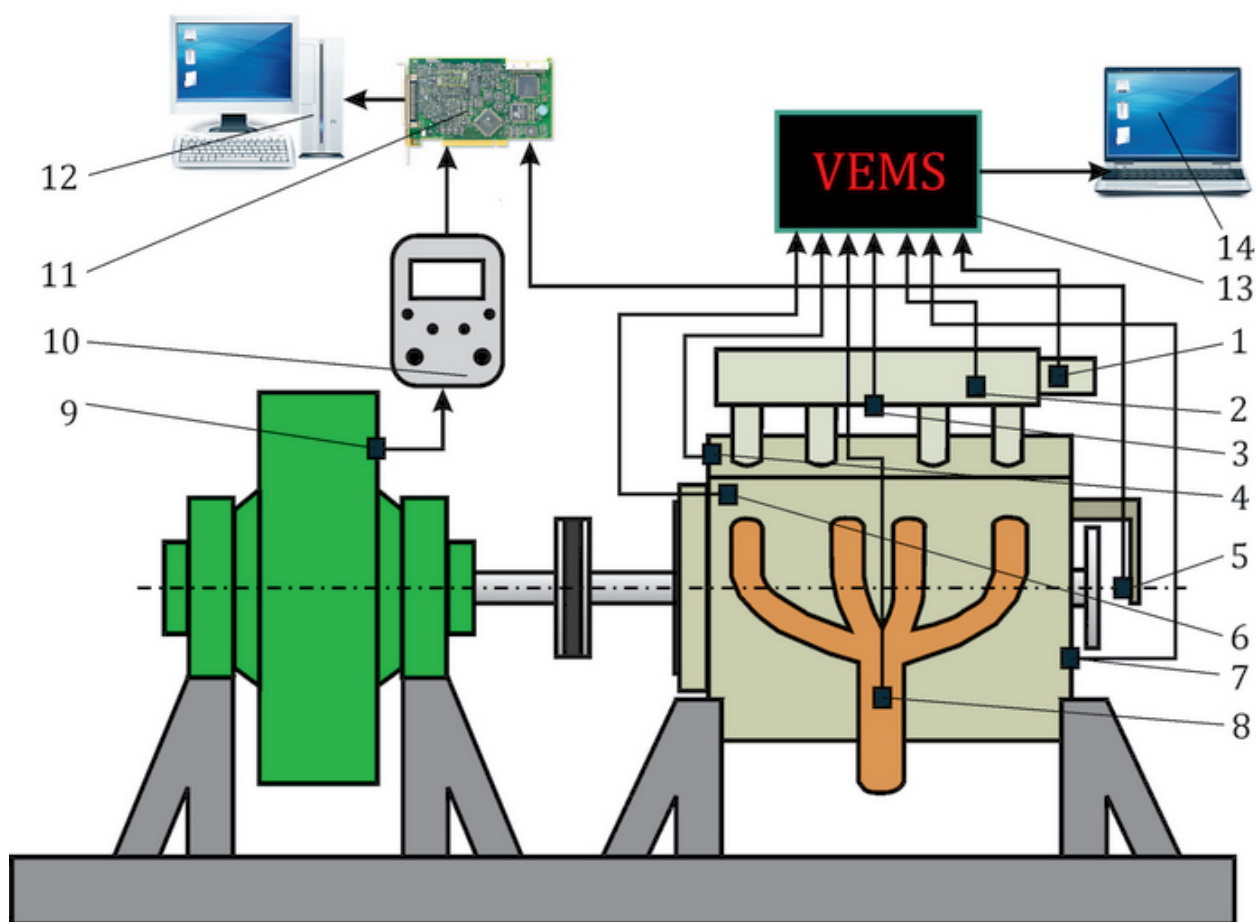
Sl. 7.1 Eksperimentalno postrojenje

Glavne komponente eksperimentalnog postrojenja su oto motor Yugo EFI 1.1 (1) i dinamometar SCHENCK W230 (2) sa svojim komandnim ormanom (3). Kolenasto vratilo motora je pomoću spojnice spregnuto sa vratilom dinamometra, što omogućava da se promenom parametara dinamometra menja spoljašnje opterećenje motora. Promena parametara dinamometra ostvaruje se pomoću odgovarajućih komandi (4) na njegovom komandnom ormanu. Upravljanje razvijenom snagom motora ostvaruje se pomoću regulacione ručice (5), koja je odgovarajućom sajлом (6) povezana sa leptirom motora.

Osim ovih, glavnih komponenti eksperimentalnog postrojenja, koje omogućavaju uspostavljanje režima rada motora odnosno izbor njegove radne tačke, tu su još i veoma važne komponente, pomoću kojih se vrši merenje relevantnih veličina i akvizicija dobijenih podataka, a koje se jednim imenom zovu merno-akvizicionom opremom.

U merno-akvizicionu opremu spadaju svi relevantni senzori, koji su uglavnom integrisani ili u okviru motora ili u okviru dinamometra, pa se ne vide na Sl. 7.1, zatim pojačavač mernih signala KWS/T-5 (7) sa obavezanim pratećim laboratorijskim izvorom jednosmernog napona EA-PS 2016-100 (8), kao i adaptivni sistem za upravljanje motorom VEMS<sup>32</sup> V3.6 (9). Po svojoj suštini, VEMS ne bi trebalo da spada u merno-akvizicionu opremu, jer je njegova primarna funkcija upravljanje motorom, kako mu i sâm naziv kaže. Međutim, u ovom istraživanju su u većoj meri korišćene pogodnosti koje VEMS nudi u prikupljanju i obradi podataka o važnim motorskim parametrima, pa je zato i uvršten u kategoriju merno-akvizicione opreme. Tu su još i dva personalna računara: laptop (10) i desktop (11), čije će uloge uskoro biti objašnjene.

Kompletniji uvid u merno-akvizicionu opremu dobija se posmatranjem Sl. 7.2, koja prikazuje tok mernih signala na eksperimentalnom postrojenju.

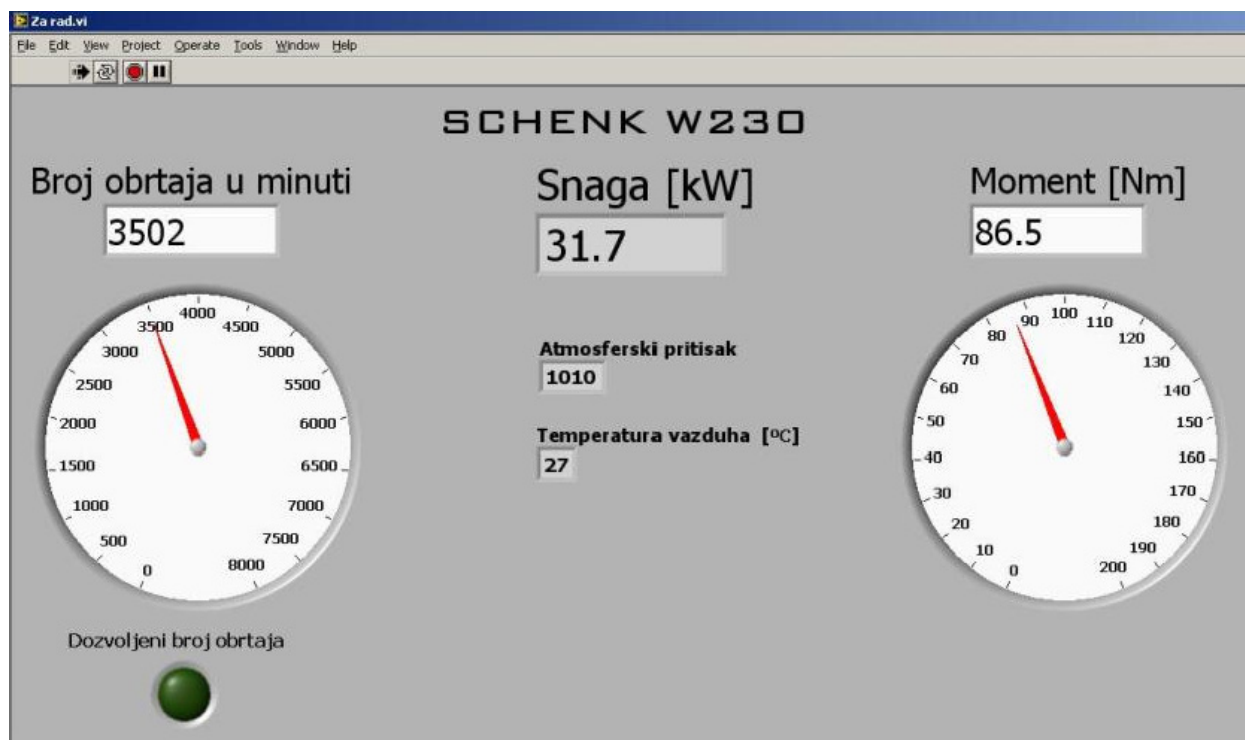


Sl. 7.2 Šematski prikaz eksperimentalnog postrojenja: 1 – senzor položaja leptira, 2 – senzor pritiska u usisnom kolektoru, 3 – senzor temperature u usisnom kolektoru, 4 – senzor referentnog položaja bregastog vratila (Hall-ov davač), 5 – enkoder, 6 – senzor temperature rashladne tečnosti, 7 – senzor broja obrtaja, 8 – širokopojasna  $\lambda$ -sonda, 9 – senzor sile, 10 – merno pojačalo, 11 – akviziciona kartica, 12 – desktop računar, 13 – VEMS upravljačka jedinica (kontroler), 14 – laptop računar

Merno-akviziciona oprema je povezana sa glavnim komponentama postrojenja tako da se omogući prikupljanje podataka za obuku i testiranje neuronske mreže, na način kako je to učinjeno i pri izvođenju virtuelnih eksperimenata, opisanih u Poglavlju 5. To znači da je u jednoj baznoj radnoj tački motora trebalo izvršiti merenje i snimanje relevantnih motorskih promenljivih za različita stanja usisnog sistema. Za potrebe praćenja radne tačke

<sup>32</sup> VEMS – Versatile Engine Management System, u prevodu: prilagodljivi sistem za upravljanje motorom

motora, korišćeni su merni signali senzora sile (9) i enkodera (5). Signal davača sile se pojačava pri prolasku kroz analogno jednocanalno merno pojačalo KWS/T-5, a zatim se vrši njegova konverzija u digitalni oblik u kartici za akviziciju podataka NI PCI-6070E (11), koja prihvata i direktni signal enkodera (5). Ova kartica je ugrađena u desktop računar (12), na kojem funkcioniše jedna LabView aplikacija, pomoću koje se mogu pratiti osnovni parametri režima rada motora. Aplikacija je razvijena za potrebe istraživanja opisanih u referenci [116], a izgled njenog korisničkog interfejsa prikazan je na Sl. 7.3



Sl. 7.3 Korisnički interfejs LabView aplikacije za praćenje osnovnih parametara režima rada motora

Dalje, kako je već rečeno, u sastavu eksperimentalnog postrojenja je još jedan personalni računar (pozicija 14, Sl. 7.2). Na ovom računaru je instaliran softver *VemsTune*, koji služi za komunikaciju sa VEMS kontrolerom (pozicija 13, Sl. 7.2). Između ostalog, VEMS kontroler prihvata merne signale sledećih relevantnih motorskih senzora, koji su označeni na Sl. 7.2: senzora položaja leptira (1), senzora pritiska u usisnom kolektoru (2), senzora temperature u usisnom kolektoru (3), senzora broja obrtaja (7), širokopojasne  $\lambda$ -sonde (8), senzora referentnog položaja bregastog vratila (4) i senzora temperature rashladne tečnosti (6). VEMS kontroler obrađuje ove signale, i na osnovu toga, sa jedne strane, generiše odgovarajuće komandne signale, pomoću kojih upravlja motorom, a sa druge strane prosleđuje rezultate obrade signala laptop računaru (14), na kojem se oni snimaju i grafički prikazuju pomoću *VemsTune* softvera. Međutim, s obzirom da je u ovom istraživanju u prvom planu akviziciona funkcija VEMS-a, a i da bi se izbeglo bespotrebno uslozňavanje šeme postrojenja, komandni signali su izostavljeni sa Sl. 7.2.

## 7.3 Opis pojedinih komponenti eksperimentalnog postrojenja

### 7.3.1 Motor

Pri izvođenju eksperimentalnog dela istraživanja korišćen je oto motor Yugo EFI 1.1 sa multipoint ubrizgavanjem goriva u usisne kanale. Sa aspekta dijagnostike usisnog sistema ovog motora, najvažnija njegova svojstva su način punjenja cilindra vazduhom i način određivanja masenog protoka tog vazduha. U pitanju je atmosferski motor, pa se cilindri pune vazduhom prirodnim putem, bez prehranjivanja. To znači da je pritisak u usisnom sistemu uvek niži od pritiska okoline i da će, pri postojanju bilo kakvog neželjenog otvora na usisnom sistemu, strujanje kroz taj otvor uvek biti u smeru prema motoru. Ovaj motor nema protokomer vazduha, tako da se maseni protok vazduha određuje indirektno, računanjem, na osnovu pritiska u usisnom kolektoru, temperature u usisnom kolektoru i broja obrtaja kolenastog vratila.

Ostali važni fabrički podaci, koji daju kompletniju sliku o ovom motoru, sažeti su u *Tab. 7.1 [117]*.

*Tab. 7.1 Osnovni tehnički podaci motora YUGO EFI 1.1*

Broj cilindara	4
Broj ventila po cilindru	2
Radna zapremina	1116 cm <sup>3</sup>
Prečnik klipa	80 mm
Hod klipa	55.5 mm
Maksimalna snaga motora	45kW pri 5600 min <sup>-1</sup>
Maksimalni obrtni moment	85Nm pri 3800 min <sup>-1</sup>
Stepen kompresije	9.2
Način hlađenja	Tečnošću

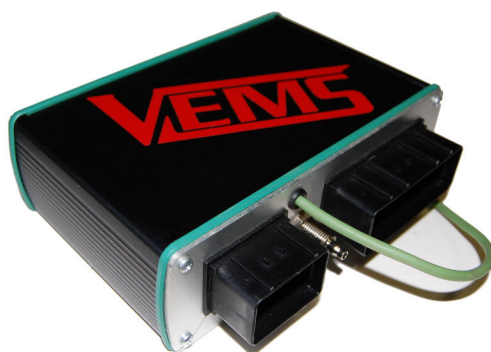
Međutim, motor koji je ispitivan na probnom stolu bio je nešto izmenjen u odnosu na svoje originalno, fabričko stanje. Najkrupnija izvršena izmena je u vezi sa sistemom za upravljanje ubrizgavanjem goriva i paljenjem smeše. Umesto originalne elektronske upravljačke jedinice Bosch-ovog sistema Motronic 3.1, ugrađena je elektronska upravljačka jedinica VEMS V3.6. Najvažniji razlog ove izmene je taj, što je VEMS, zajedno sa svojim pratećim aplikativnim softverom *VemsTune*, veoma pogodan za prikupljanje podataka koji su neophodni za razvoj laboratorijskog dijagnostičkog modela. Ovde će biti nabrojane najvažnije prednosti primene VEMS-a, posmatrano sa stanovišta razvoja planiranog dijagnostičkog modela:

- Pomoću VEMS-a se može promeniti ofset senzora pritiska u usisnom kolektoru, što se može iskoristiti za izazivanje neispravnosti očitavanja ovog senzora.
- Softver „VemsTune“ omogućava da se u realnom vremenu prati veliki broj izmerenih i izračunatih parametara motora, kao što su: otvor leptira, broj obrtaja motora, pritisak u usisnom kolektoru, temperatura u usisnom kolektoru, temperatura rashladne tečnosti, koeficijent viška vazduha, dužina impulsa ubrizgavanja, ugao pretpaljenja itd.
- „VemsTune“ omogućava da se promene motorskih parametara, koje su praćene u proizvoljnom vremenskom intervalu, zapisuju u odgovarajući log-fajl, i da se na taj način kreira svojevrsan „istorijat“ rada motora.

Pored nabrojanih pogodnosti, VEMS nudi gotovo beskonačan broj mogućnosti kada je u pitanju rekonfigurisanje velikog broja mapa podataka, kojima se na najfiniji način može podešavati ponašanje motora u njegovom celokupnom radnom opsegu. Međutim, tokom

ovog istraživanja pomenute opcije VEMS-a nisu u značajnijoj meri eksploatisane, a zainteresovani čitalac se sa njima može bliže upoznati u referenci [118].

Izgled hardverske komponente VEMS-a dat je na Sl. 7.4, a korisnički interfejs pripadajućeg softvera „VemsTune“, tokom izvođenja merenja u jednoj baznoj radnoj tački motora, prikazan je na Sl. 7.5.



Sl. 7.4 VEMS upravljačka jedinica (VEMS kontroler)



Sl. 7.5 Korisnički interfejs „VemsTune“-a

Prilikom izvođenja eksperimentalnog dela ovog istraživanja, korišćeni su signali određenog broja originalnih senzora, integrisanih u okviru samog motora, pa će i oni biti pobrojani ovde, u odeljku koji opisuje motor. To su: senzor položaja leptira, senzor broja obrtaja motora, senzor temperature u usisnom kolektoru i senzor temperature rashladne tečnosti. Svi oni su prikazani na Sl. 7.6.





(a)



(b)



(c)



(d)

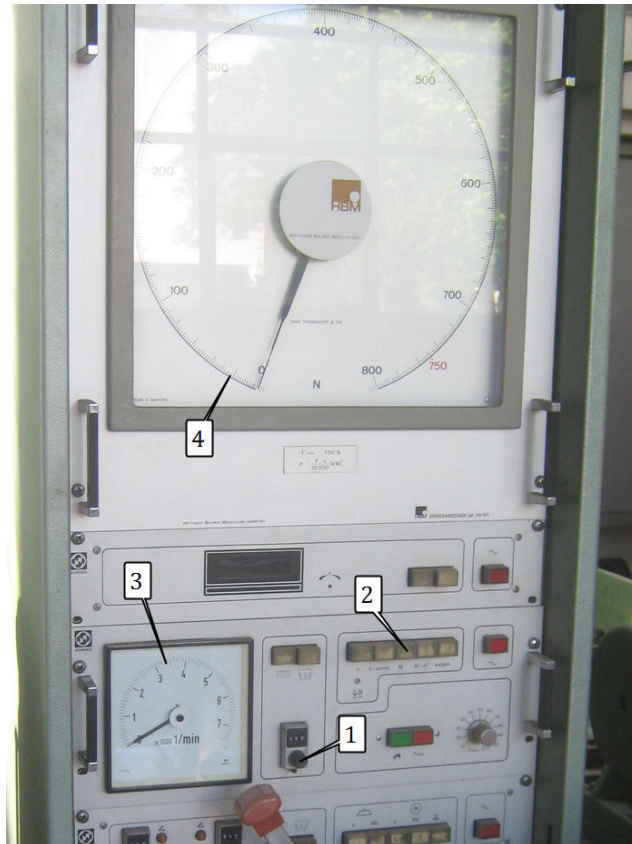
Sl. 7.6 Originalni senzori na motoru EFI 1.1: (a) senzor položaja leptira, (b) senzor broja obrtaja motora, (c) senzor temperature u usisnom kolektoru, (d) senzor temperature rashladne tečnosti

### 7.3.2 Dinamometar SCHENCK W230

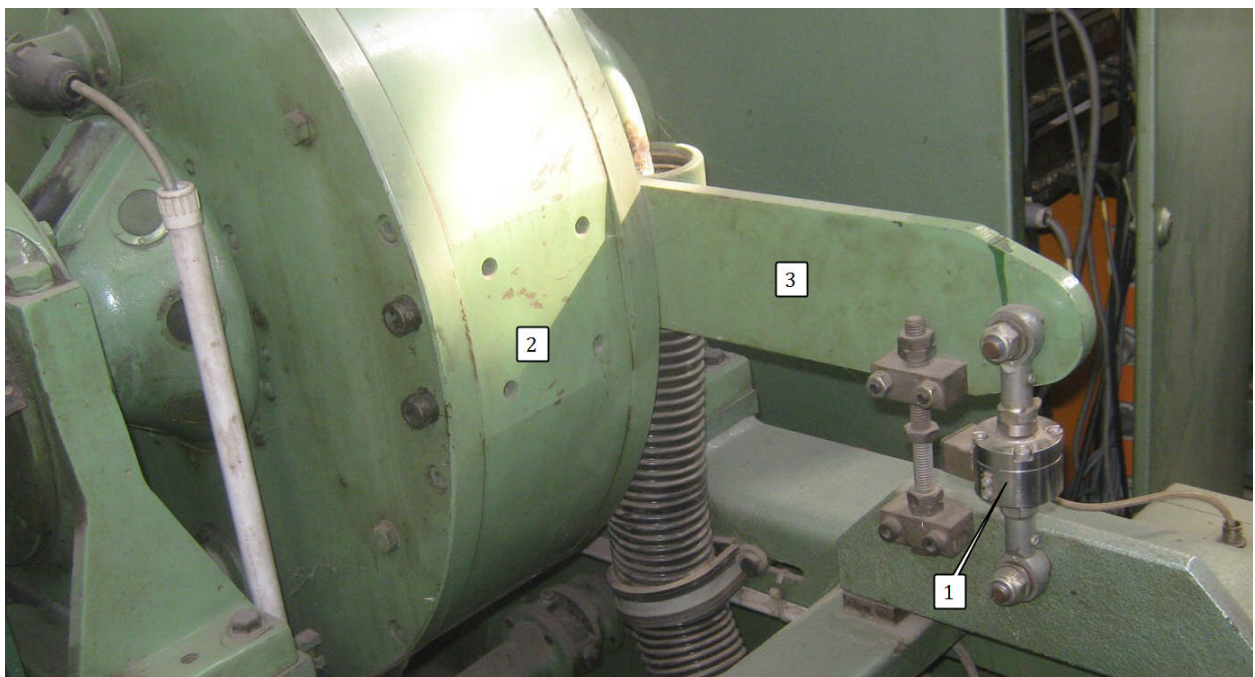
Kako se vidi na Sl. 7.1, ispitivani motor je spregnut sa dinamometrom SCHENCK W230. Ova sprega je ostvarena tako što su im vratila međusobno povezana elastičnom spojnicom, kako bi se kompenzovala njihova eventualna nesaosnost. Na taj način je omogućeno da se vratilo rotora dinamometra obrće zajedno sa kolenastim vratilom motora. Kada je dinamometar aktivan, kroz namotaje rotora protiče električna struja, usled čijeg elektromagnetnog polja se u namotajima statora indukuju tzv. vihorne struje. Pojavom ovih vihoranih struja generiše se i elektromagnetno polje statora. Uzajamnom interakcijom ova dva elektromagnetna polja dobija se efekat kočenja motora SUS, pri čemu se intenzitet kočenja može regulisati promenom jačine električne struje rotora. Promena intenziteta kočenja motora ostvaruje se pomoću odgovarajuće komande (točkić potenciometra, Sl. 7.7, pozicija 1) na komandnom ormanu dinamometra. Osim ove osnovne komande, veoma su važne i komande kojima se može zadati željeno ponašanje sistema *motor-dinamometar*, pri nekom poremećaju koji dovodi do promene radne tačke motora. Pomenute komande su označene pozicijom 2 na Sl. 7.7 i naročito su bitne za potrebe istraživanja u ovom radu, kada se izazivanjem određenih neispravnosti u usisnom sistemu, motor izvodi iz trenutne radne tačke. Na primer, dinamometar SCHENCK W230 nudi mogućnost, da se pri nekom poremećaju broj obrtaja ili moment kočenja zadrže na nivou pre poremećaja, ili da vrednost momenta kočenja bude proporcionalna kvadratu broja obrtaja.

Osnovna namena dinamometra SCHENCK W230 je određivanje obrtnog momenta motora, iako mu je direktni zadatak, kako mu ime kaže, merenje sile. Merenje sile se ostvaruje pomoću senzora sile HBM U2, čiji je merni opseg od 0 do 1960N, a klasa tačnosti 0.5%. Ovaj senzor je sastavni deo dinamometra, na koji je postavljen tako, da mu je osa na tačno određenom normalnom rastojanju od ose obrtanja rotora dinamometra, što je prikazano na Sl. 7.8. Meri se sila, kojom stator dinamometra (2), preko odgovarajuće poluge

(3), deluje na senzor sile (1). Intenzitet izmerene sile, koji se registruje na odgovarajućoj skali (pozicija 4, Sl. 7.7), uz poznatu dužinu kraka sile, definiše moment kočenja motora. Na ustaljenom režimu rada, vrednost ovog momenta jednaka je obrtnom momentu motora, tako da je on na taj način indirektno određen.



Sl. 7.7 Komandni orman sa najvažnijim detaljima: 1) točkić za podešavanje intenziteta kočenja, 2) tasteri za izbor režima kočenja, 3) skala broja obrtaja, 4) skala sile kočenja



Sl. 7.8 Princip određivanja obrtnog momenta motora pomoću dinamometru SCHENCK W230: 1) senzor sile HBM U2, 2) stator dinamomentra 3) poluga statora

Neke osnovne karakteristike dinamometra SCHENCK W230 date su u *Tab. 7.2*, a kompletna tehnička specifikacija ovog uređaja može se naći u referenci [119].

*Tab. 7.2 Osnovni tehnički podaci dinamometra SCHENCK W230*

Maksimalna snaga	230 kW
Maksimalni obrtni moment	750 Nm
Minimalni broj obrtaja	620 min <sup>-1</sup>
Maksimalni broj obrtaja	7500 min <sup>-1</sup>
Minimalni broj obrtaja pri maksimalnoj snazi	2928 min <sup>-1</sup>
Moment inercije	0.53 kgm <sup>2</sup>
Radna temperatura	od 0 do 70 °C bez aditiva, a -25 do 70 °C sa antifrizom i redukovanom snagom od 10 do 20%

### 7.3.3 Ostala merno-akviziciona oprema

Kratki prikaz osnovnih karakteristika ostalih korišćenih uređaja, svrstanih u merno-akvizicionu opremu, daje se u nastavku. Osim pojačavača mernih signala i kartice za akviziciju podataka, ovde su dati i kratki opisi motorskih senzora, kojih nema u fabričkoj izvedbi motora Yugo EFI 1.1, ali su postavljeni na motor za potrebe ovog istraživanja ili su nasleđeni iz eksperimentalnog postrojenja opisanog u referenci [116].

#### Pojačavač mernih signala KWS/T-5

Ovo je jednokanalni uređaj koji se koristi za pojačavanje mernih signala za senzore na bazi mernih traka, potencijometarske senzore i induktivne senzore. Ima analogni izlaz i klasu tačnosti 1%. Zahteva stabilan jednosmerni napon, zbog čega je korišćen u sprezi sa laboratorijskim napajanjem EA-PS 2016-100, čije su osnovne tehničke karakteristike:

- jednosmerni napon od 0 do 16V,
- jačina struje od 0 do 10A,
- stabilnost pri opterećenju 0-100%: <100 mV, <10mA

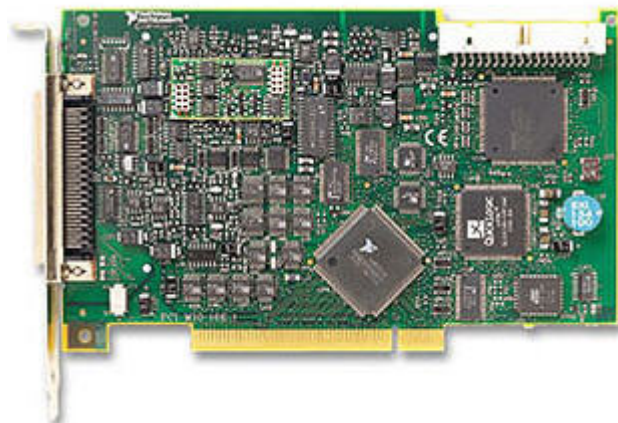
*Sl. 7.9* prikazuje izgled pojačavača (levo) i njegovog napajanja (desno).



*Sl. 7.9 Merno pojačalo KWS /T-5 sa napajanjem EA-PS 2016-100*

### Kartica za akviziciju podataka NI PCI-6070E

Ovaj uređaj vrši obradu signala, prvenstveno konverziju analognih u digitalne, ako je potrebno, i tako ih transformiše u oblik koji je pogodan za manipulisanje u personalnom računaru. Ova akviziciona kartica je interni uređaj personalnog računara, što znači da se ugrađuje na njegovu matičnu ploču, kao i bilo koja druga kartica. Njen izgled je prikazan na *Sl. 7.10*, a osnovne tehničke karakteristike su date u *Tab. 7.3* [116].



*Sl. 7.10 Izgled akvizicione kartice NI PCI-6070E*

*Tab. 7.3 Tehničke karakteristike kartice NI PCI-6070E*

Analogni ulaz	
Broj kanala	16
Rezolucija	12 bit
Brzina uzorkovanja	1.25 MS/s
Maksimalni opseg	od -10V do +10V
Minimalni opseg	od -50mV do +50mV
Analogni izlaz	
Broj kanala	2
Rezolucija	12 bit
Maksimalni opseg	od -10V do +10V
Digitalni ulaz/izlaz	
Broj kanala	8
Opseg ulaznog i izlaznog napona	od 0V do +5V

### Inkrementalni rotacioni enkoder HEDSS ISC 3806

Ovaj optički uređaj je nasleđen iz postrojenja, opisanog u referenci [116], gde je korišćen za precizno određivanje položaja kolenastog vratila pri indiciranju motora. Ovde nije bilo neophodno tako precizno određivanje položaja kolenastog vratila, ali je enkoder ostao sastavni deo i eksperimentalnog postrojenja koje je korišćeno u ovom istraživanju. Enkoder HEDSS ISC3806 generiše 2000 impulsa po jednom obrtaju i šalje ih direktno u karticu za akviziciju podataka. To je iskorišćeno za određivanje broja obrtaja motora u ranije pomenutoj LabView aplikaciji koja komunicira sa tom karticom, i koja služi za praćenje trenutne radne tačke motora. Dakle, u ovom istraživanju enkoder je upotrebljen u funkciji senzora broja obrtaja. Njegove osnovne tehničke karakteristike date su u *Tab. 7.4*, a više detalja je dostupno u [120]. Fotografija montiranog uređaja prikazana je na *Sl. 7.11*.

Tab. 7.4 Osnovne tehničke karakteristike enkodera HEDSS ISC 3806

Broj impulsa	2000 po obrtaju
Maksimalni broj obrtaja	5000 min <sup>-1</sup>
Napajanje	5-24 V
Potrošnja struje	≤ 120 mA
Radna temperatura	približno 25°C - 85°C
Moment inercije rotora	oko 0.00035 kgm <sup>2</sup>
Masa	oko 130 g (sa 0.5m kabla)



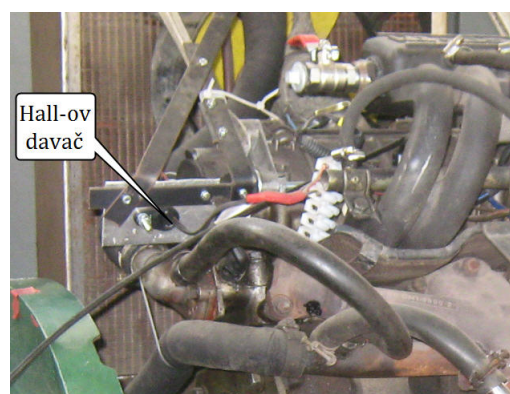
Sl. 7.11 Izgled montiranog enkodera HEDSS ISC 3806

### Senzor referentnog položaja bregastog vratila Honeywell 1GT101DC

Ovaj senzor, koji je još poznat pod nazivom Hall-ov davač, takođe je nasleđen iz prethodno razvijenog eksperimentalnog postrojenja [116]. Svojim signalom, koji šalje u VEMS kontroler, omogućava izvođenje sekvencijalnog ubrizgavanja goriva kod ispitivanog motora. Ovo nije od primarnog značaja za istraživanje, ali je to ozbiljna promena u odnosu na istovremeno ubrizgavanje goriva, koje je izvođeno u okviru izvornog sistema Motronic 3.1. Zato je i Hall-ov davač morao biti pomenut u okviru ove tačke. Ovaj davač može da radi u temperaturnom opsegu od -40°C do +150°C, a za funkcionisanje mu je potrebna struja jačine 10mA i napon od 4,5 do 24V. Njegov izgled je prikazan na Sl. 7.12a, a njegov položaj na motoru na Sl. 7.12b.



(a)



(b)

Sl. 7.12 Hall-ov davač Honeywell 1GT101DC: (a) izgled, (b) montiran na motoru

### Senzor pritiska u usisnom kolektoru VEMS MPXH6400

Ovaj senzor služi za određivanje apsolutnog pritiska u usisnom kolektoru i svoj signal šalje VEMS kontroleru. U fabričkoj izvedbi motora YUGO EFI 1.1 postojao je senzor apsolutnog pritiska, ali je bio integrisan u elektronskoj upravljačkoj jedinici, koja je, kao što

je već rečeno, uklonjena i zamenjena VEMS kontrolerom. Zbog toga je bilo neophodno ugraditi novi senzor, a izbor je pao na model VEMS MPXH6400, čiji je izgled, u montiranom stanju na motoru, prikazan na Sl. 7.13.



Sl. 7.13 Senzor apsolutnog pritiska u usisnom kolektoru VEMS MPXH6400

Za ovo istraživanje je veoma bitno, da se pomoću softvera *VemsTune* može menjati ofset ovog senzora, što dovodi do pogrešnog očitavanja vrednosti pritiska u usisnom kolektoru. Variranjem vrednosti ofseta, mogu se izazivati različiti intenziteti pomenute neispravnosti u usisnom sistemu.

Pomoću ovog senzora može se meriti i apsolutni pritisak u usisnom sistemu nekih prehranjivanih motora, jer mu je merni opseg od 20 kPa do 250 kPa, pri čemu daje analogni izlazni naponski signal od 0.2 V do 4.8 V, respektivno. Vreme odziva mu je 0.1 ms. Više detalja o tehničkim karakteristikama ovog senzora može se naći u referenci [121].

### Širokopojasna lambda-sonda Bosch LSU 4.2

Najpre bi trebalo objasniti po čemu se širokopojasna lambda-sonda razlikuje od konvencionalne (uskopojasne).

Konvencionalna lambda-sonda deluje kao prekidač, koji šalje elektronskoj upravljačkoj jedinici motora informaciju o tome, da li je smeša bogatija ili siromašnija od željene. Međutim, izostaje informacija u kojoj meri je stvarna smeša bogatija ili siromašnija od željene. Za razliku od konvencionalne, širokopojasna lambda-sonda, u sprezi sa VEMS-om, obezbeđuje precizno očitavanje vrednosti koeficijenta viška vazduha i u zoni siromašne i u zoni bogate smeše, tako da se na korisničkom interfejsu *VemsTune*-a, može pratiti njegova trenutna vrednost tokom rada motora. Ovo dalje omogućava da se tokom izvođenja merenja ustanovi, da li se ustalila vrednost koeficijenta viška vazduha posle pojave nekog poremećaja, što je veoma bitno sa stanovišta ovog istraživanja, u kojem je jedna od osnovnih pretpostavki ustaljeni rad motora.

Još jedna bitna stvar u vezi sa širokopojasnom lambda-sondom je ta, da motori koji imaju takvu lambda-sondu mogu da rade u zatvorenoj sprezi sa njome i u oblasti siromašne i u oblasti bogate smeše. Time se povećava broj režima rada motora sa preciznijim doziranjem goriva.

Širokopojasna lambda-sonda Bosch LSU 4.2 može trajno da izdrži temperaturu 930°C, a kratkotrajno i do 1030°C. Ostale važne tehničke karakteristike ove lambda-sonde date su u Tab. 7.5 [122], a njena fotografija prikazana je na Sl. 7.14.

Tab. 7.5 Osnovne tehničke karakteristike lambda-sonde Bosch LSU 4.2

Merni opseg	Od 0.65 do $\infty$
Tačnost pri $\lambda = 0.8$	$\pm 1\%$
Tačnost pri $\lambda = 1$	$\pm 0.8\%$
Tačnost pri $\lambda = 1.7$	$\pm 5\%$
Izlazni signal	od -1.85 mA do 2.55 mA



Sl. 7.14 Širokopojasna  $\lambda$ -sonda Bosch LSU 4.2 montirana na motor Yugo EFI 1.1

## 7.4 Realizacija eksperimentalnog dela istraživanja

Kako je ranije rečeno, pri izvođenju stvarnih laboratorijskih eksperimenata, plan je bio da se to uradi tako, da što više bude po uzoru na virtuelne eksperimente, koji su opisani u Poglavlju 5. Ovde se, pre svega, misli na neispravnosti usisnih sistema virtuelnog i stvarnog motora, kao i na njihove bazne radne tačke, u kojima se te neispravnosti izazivaju. Međutim, objektivne okolnosti, navedene u Odeljku 7.1, nametnule su neka ograničenja kod laboratorijskih eksperimenata, pa su se ipak pojavile приметne razlike u odnosu na virtuelne eksperimente, u smislu manjeg broja posmatranih neispravnosti i manjeg broja baznih radnih tačaka motora.

### 7.4.1 Opis izazivanja neispravnosti u usisnom sistemu

Stanja usisnog sistema, koja su posmatrana tokom izvođenja laboratorijskih eksperimenata na motoru Yugo EFI 1.1, u cilju prikupljanja podataka za obuku neuronske mreže, navedena su u *Tab. 7.6*. Još neka stanja su uzeta u obzir prilikom prikupljanja podataka za testiranje neuronske mreže, ali o tome će biti reči kasnije, kada se bude govorilo o testiranju.

Tab. 7.6 Oznake i nazivi razmatranih stanja usisnog sistema motora Yugo EFI 1.1

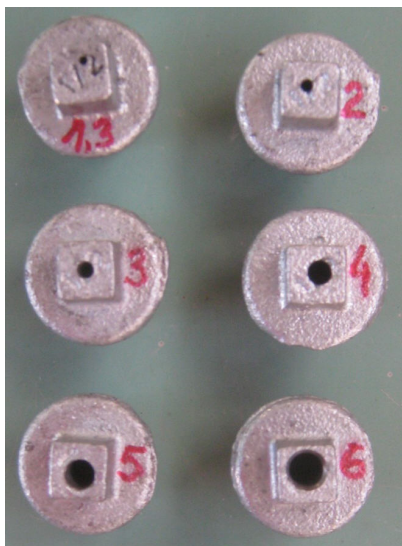
Oznaka stanja usisnog sistema	Naziv stanja usisnog sistema
Sve OK	Nema neispravnosti
FaV 2mm	Ulaz „falš“-vazduha kroz otvor prečnika 2mm
FaV 4mm	Ulaz „falš“-vazduha kroz otvor prečnika 4mm
FaV 6mm	Ulaz „falš“-vazduha kroz otvor prečnika 6mm
MAP -10	Senzor pritiska u usisnom kolektoru (MAP senzor) očitava za 10 kPa nižu vrednost
MAP +10	Senzor pritiska u usisnom kolektoru (MAP senzor) očitava za 10 kPa višu vrednost
MAP +20	Senzor pritiska u usisnom kolektoru (MAP senzor) očitava za 20 kPa višu vrednost

Poređenjem stanja usisnog sistema iz *Tab. 7.6* sa onima, datim u *Tab. 5.6*, zapaža se da pri laboratorijskim eksperimentima nisu razmatrane neispravnosti senzora temperature u usisnom sistemu i neispravnosti EGR sistema. Razlozi za izostavljanje oba navedena tipa neispravnosti iz razmatranja su tehničke prirode. Naime, ispitivani motor nije posedovao EGR sistem, pa su neispravnosti koje se odnose na njega automatski eliminisane iz

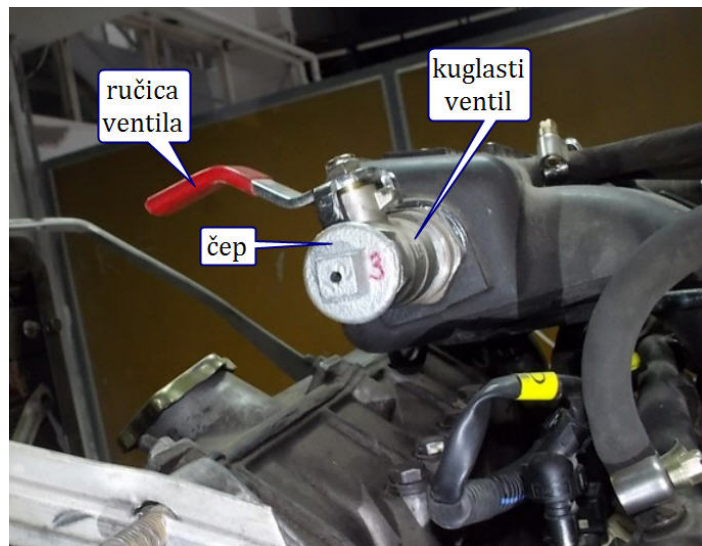
razmatranja. Što se tiče neispravnosti senzora temperature, odustalo se pre svega zato, da bi se smanjio broj neispravnosti kojima je ispitivani motor podvrgavan, jer je postojao strah da motor neće izdržati do kraja istraživanja, ako se tako frekventno i tako dugo bude primoravao da radi u neregularnim režimima. Da su uzete u razmatranje greške očitavanja temperature u usisnom kolektoru, moralo bi se iz predostrožnosti odustati od neispravnosti senzora pritiska ili „falš“-vazduha, ali je procenjeno da ove dve neispravnosti imaju ozbiljniji uticaj na rad motora, pa su greške senzora temperature ostavljene za neko buduće istraživanje.

Imajući u vidu sve navedeno, preostale su neispravnosti tipa „ulazak falš-vazduha u usisni sistem“ i „pogrešno očitavanje senzora pritiska u usisnom kolektoru“. Za praktičnu realizaciju ovih neispravnosti tokom ispitivanja motora osmišljeni su načini, koji će uskoro biti predstavljeni. Pri tome je bilo veoma bitno da motor vrlo brzo može da pređe iz ispravnog u neispravno stanje i obrnuto. Ovaj kriterijum naročito dolazi do izražaja kada treba izvesti veliki broj merenja, a to je bio slučaj u ovom istraživanju.

Kada je reč o ulazu „falš“-vazduha u usisni sistem, izazivanje ove neispravnosti je omogućeno montiranjem kuglastog ventila 1/2" sa ručicom na zadnji deo usisnog kolektora. Slobodan kraj kuglastog ventila služio je za postavljanje vodovodnih čepova sa probušenim otvorima različitih prečnika, u zavisnosti od toga, koja konkretna neispravnost se želi izazvati. Na Sl. 7.15a prikazani su čepovi koji su korišćeni u preliminarnim i glavnim merenjima, a na Sl. 7.15b, prikazan je montirani kuglasti ventil sa jednim od čepova.



(a)



(b)

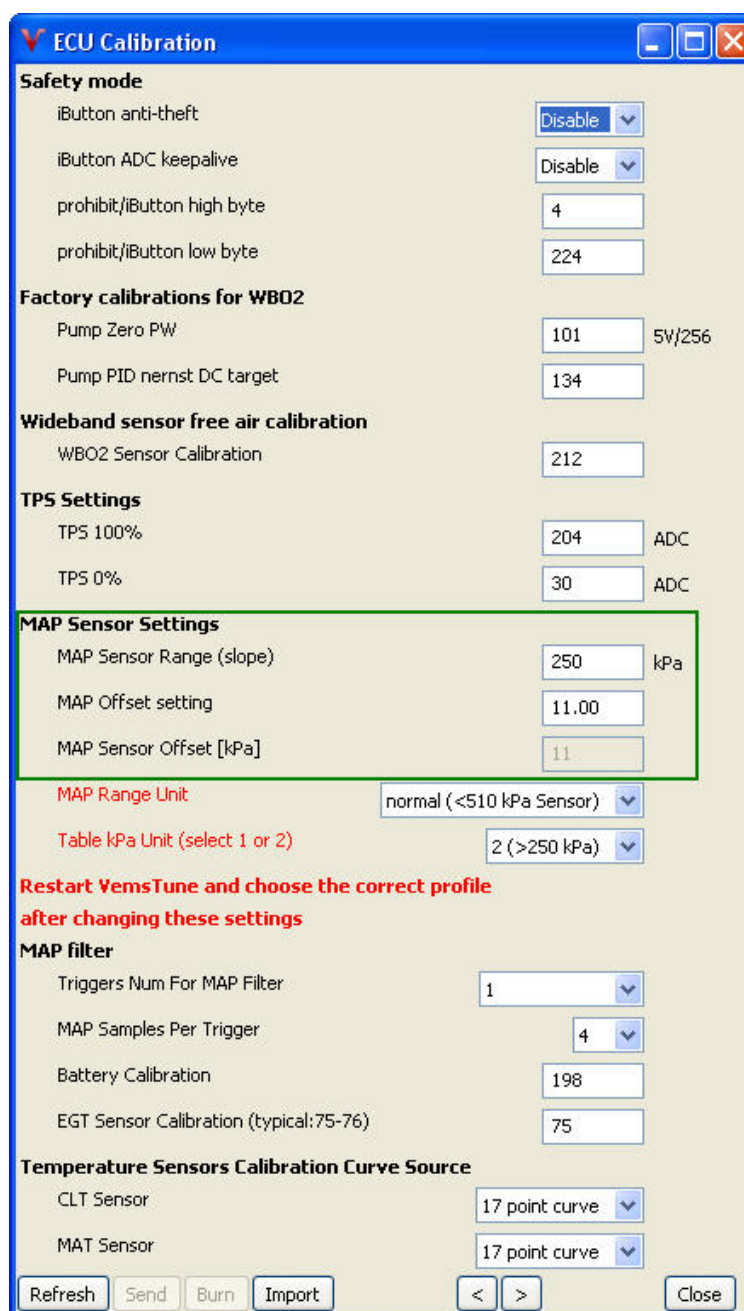
Sl. 7.15 Komponente za izazivanje ulaza „falš“-vazduha: (a) čepovi sa probušenim otvorima, (b) montirani kuglasti ventil sa jednim od čepova

Pomoću ručice kuglastog ventila veoma se lako može omogućiti ili onemogućiti ulaz „falš“-vazduha u usisni kolektor, čime je ispunjen postavljeni kriterijum brzog prelaska motora u toku rada iz ispravnog u neispravno stanje i obrnuto.

Što se tiče pogrešnog očitavanja vrednosti pritiska u usisnom kolektoru, ta neispravnost je tokom izvođenja eksperimenata izazivana softverskim putem. Već je ranije pomenuto da u softveru *VemsTune* postoji opcija promene vrednosti ofseta davača pritiska u usisnom kolektoru. Na Sl. 7.16 je prikazan prozor za kalibrisanje VEMS upravljačke jedinice, koji je korišćen za te svrhe, a u njemu je uokviren deo koji se odnosi na podešavanja davača pritiska. Pre nego što su započeta bilo kakva merenja motorskih parametara, izvršena je provera podrazumevanog, fabričkog ofseta na atmosferskom pritisku i ustanovljeno je da nije potrebno dodatno kalibrisanje. Fabrička vrednost ofseta je 11kPa, a povećanjem ove



vrednosti na 21kPa i 31kPa i njenim smanjenjem na 1kPa, izazivane su neispravnosti „MAP+10“, „MAP+20“ i „MAP-10“, respektivno.



Sl. 7.16 Ilustracija prozora u VemsTune-u, u kojem se može promeniti ofset MAP senzora

Pomoću prikazanog prozora se veoma brzo može tokom rada menjati stanje motora iz ispravnog u neispravno i obrnuto, čime je zadovoljen ranije postavljeni kriterijum. Jednostavno se pozove ovaj prozor, unese se željena vrednost ofseta davača pritiska i posle klika na dugme „send“, pa „burn“, davač već očitava pogrešnu vrednost.

Za svih sedam stanja, pobrojanih u Tab. 7.6, vršena su merenja pri radu motora u tačno određenim baznim radnim tačkama i istorijat promene merenih veličina sniman je u takozvani log-fajl pomoću softvera VemsTune. Pod radom motora u jednoj baznoj radnoj tački ovde se podrazumeva ustaljeni rad motora na određenom broju obrtaja, pri određenom momentu kočenja. Pre nego što bude opisan postupak snimanja podataka za jedno stanje usisnog sistema u jednoj baznoj radnoj tački, biće prikazan način na koji su određene sve bazne radne tačke, u kojima su vršena merenja.

## 7.4.2 Određivanje baznih radnih tačaka motora

Bazne radne tačke u kojima će motor ustaljeno raditi prilikom izvođenja merenja, određene su kao presečne tačke parcijalnih brzinskih karakteristika obrtnog momenta motora i brzinskih karakteristika momenta otpora.

Parcijalne brzinske karakteristike obrtnog momenta motora su snimane na pet različitih brojeva obrtaja: 1500, 2000, 2500, 3000 i 3500  $\text{min}^{-1}$ . Pri snimanju ovih karakteristika, iskorišćena je opcija kočenja pri konstantnom broju obrtaja, koju nudi dinamometar SCHENCK W230. Tako je pri svakom broju obrtaja izvršeno određivanje obrtnog momenta motora za 15 različitih otvora leptira: 5%, 8%, 10%, 12%, 15%, 17%, 18%, 19%, 20%, 25%, 30%, 35%, 40%, 45% i 50%. Celokupni postupak je ponovljen tri puta u dva različita dana i svaki put su dobijeni približno isti rezultati, koji su prikazani u *Tab. 7.7*.

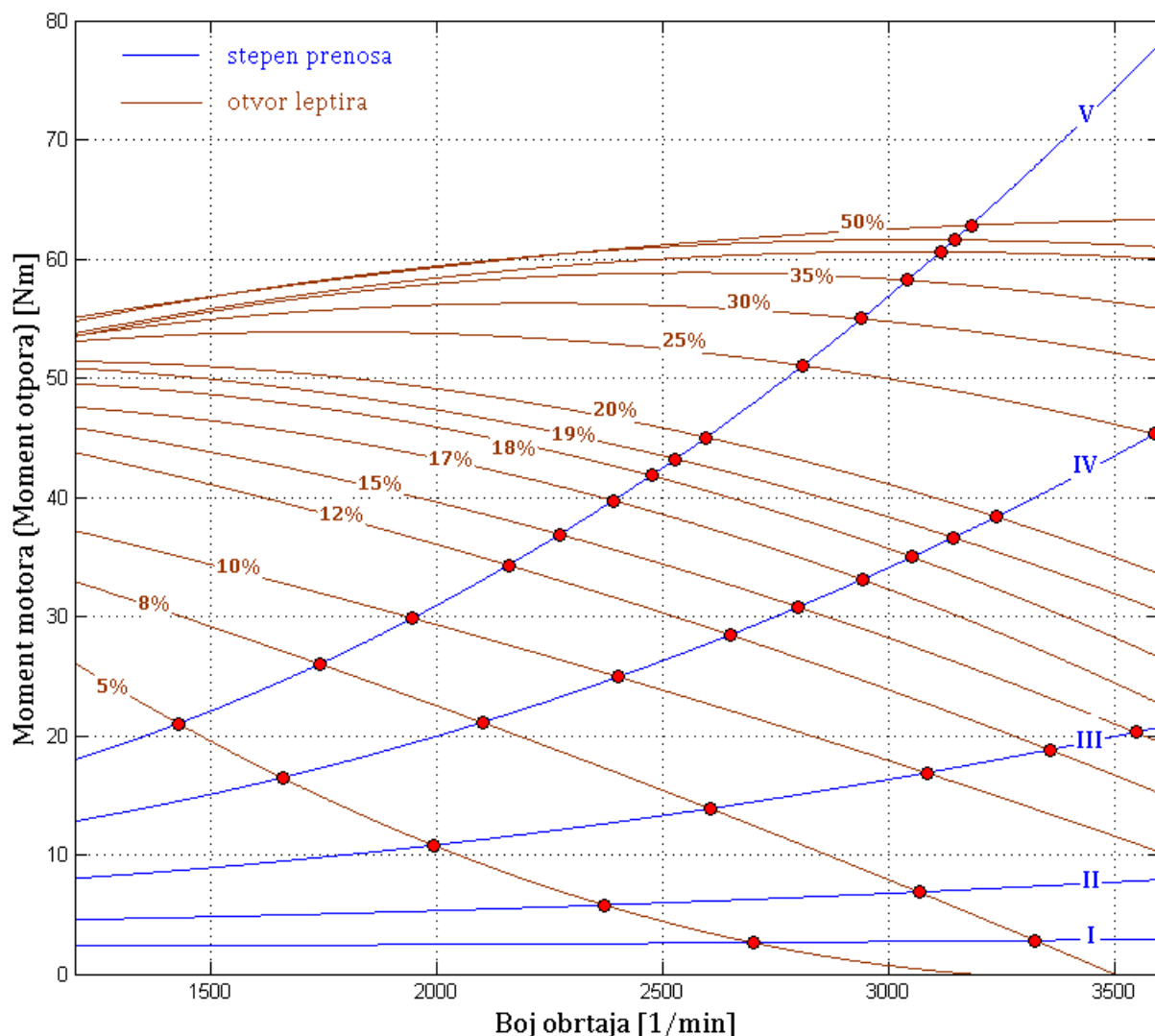
*Tab. 7.7 Rezultati određivanja parcijalnih brzinskih karakteristika momenta ispitivanog motora*

Vrednosti obrtnog momenta motora [Nm]					
Broj obrtaja [ $\text{min}^{-1}$ ]	1500	2000	2500	3000	3500
Otvor leptira [%]	1500	2000	2500	3000	3500
5	19	12	4	0	0
8	29	23	15	8	0
10	34	30	24	17	12
12	41	36	31	23	17
15	44	39	35	28	21
17	47	42	39	33	24
18	49	45	42	36	28
19	50	47	44	38	32
20	51	49	46	41	35
25	54	53	53	50	46
30	55	56	56	55	52
35	56	57	59	59	56
40	56	58	60	61	60
45	57	59	61	62	61
50	57	59	61	63	63

Za određivanje brzinske karakteristike momenta otpora upotrebljeni su isti podaci (*Tab. 5.1* na strani 79) koji su u simulatoru neispravnosti korišćeni za određivanje snage potrebne za savladavanje otpora na zamajcu, pri različitim stepenima prenosa menjačkog prenosnika vozila. To znači da je pretpostavljeno da ispitivani motor pogoni isto vozilo i pod istim uslovima, kao i motor korišćen u virtuelnim eksperimentima. Vrednosti momenta otpora pri različitim brojevima obrtaja određene su na osnovu već pomenute snage potrebne za savladavanje otpora na zamajcu i odgovarajuće ugaone brzine kolenastog vratila motora, kombinovanjem jednačina (5.7) i (5.8). Na taj način dobijene su brzinske karakteristike momenta otpora na zamajcu u različitim stepenima prenosa za posmatrano vozilo.

U cilju određivanja baznih radnih tačaka ispitivanog motora, pod opisanim uslovima, napisan je namenski Matlab program pod nazivom *radneTackeMotora*. Ovaj program učitava iz Excel-fajla podatke prikazane u *Tab. 7.7*, posle čega vrši njihovo fitovanje i iscrtavanje krivih obrtnog momenta motora. Zatim izračunava i iscrtava brzinske karakteristike momenta otpora na osnovu relevantnih podataka koji su integrisani u samom programu. Potom određuje presečne tačke između ove dve grupe krivih i obezbeđuje da se koordinate svake od tih tačaka (obrotni moment i broj obrtaja motora) snime u jedan Excel-fajl. Program *radneTackeMotora* ovo može da uradi za bilo koji motor,

ukoliko su mu za taj motor na raspolaganju Excel-fajl sa podacima dobijenim pri snimanju parcijalnih brzinskih karakteristika obrtnog momenta i podaci potrebni za određivanje momenta otpora na zamajcu. Izvorni kod programa dat je u prilogu P5.1, a rezultati njegove primene na ispitivani motor prikazani su na *Sl. 7.17*.



*Sl. 7.17* Bazne radne tačke motora Yugo EFI 1.1, za zadate podatke o otporima koje treba da savlada, u različitim stepenima prenosa

Plavim linijama su prikazane brzinske karakteristike momenta otpora na zamajcu, pri različitim stepenima prenosa, a tamno-crvenim, parcijalne brzinske karakteristike obrtnog momenta motora za 15 različitih položaja leptira. Preseci ove dve grupe krivih označeni su crvenim tačkama, kojih na konkretnom dijagramu ima 34. Iz ranije objašnjenih razloga, nije bilo moguće da se u svim ovim tačkama izazivaju neispravnosti iz *Tab. 7.7*. Zato je odlučeno da se merenja relevantnih motorskih parametara ograniče na jedan stepen prenosa i da se, u skladu sa tim, razvije samo jedna neuronska mreža. Izbor je pao na peti stepen prenosa, jer na toj krivoj ima najviše presečnih tačaka, od kojih se mogu izabrati neke, u kojima će se izazivati neispravnosti. U *Tab. 7.8* prikazane su odabrane bazne radne tačke, a može se primetiti da su one veoma zgusnute od otvora leptira 15% do otvora leptira 20%. Ovo je učinjeno sa namerom da se neke od njih upotrebe samo zbog prikupljanja podataka za testiranje neuronske mreže, dok će u ostalima biti izvedena merenja i u cilju obuke i u cilju testiranja neuronske mreže. Konkretno, bazne radne tačke koje odgovaraju otvorima leptira 18% i 19% planirane su za prikupljanje podataka koji će služiti samo za testiranje neuronske mreže.

Tab. 7.8 Bazne radne tačke ispitivanog motora, u kojima će se izazivati neispravnosti

Otvor leptira [%]	Obrtni moment [Nm]	Broj obrtaja [min <sup>-1</sup> ]
8	26	1742
10	30	1948
12	34	2160
15	37	2273
17	40	2392
18	42	2475
19	43	2529
20	45	2597

### 7.4.3 Postupak snimanja podataka za jedno stanje usisnog sistema

Osnovni preduslov za početak snimanja podataka bio je da ispitivani motor radi bez neispravnosti i da bude zagrejan na radnu temperaturu. To znači, da nikakva merenja nisu vršena, ukoliko je temperatura rashladne tečnosti motora bila ispod 80°C. Tokom izvođenja svih merenja, praćena je vrednost ove temperature i, ako se uzmu u obzir sva izvršena merenja tokom istraživanja, ona se kretala između 80°C i 87°C. Naravno, tokom jednog merenja, temperatura rashladne tečnosti je uvek imala konstantnu vrednost iz pomenutog intervala.

Pošto se ustanovi da je temperatura rashladne tečnosti stabilna, potrebno je dovesti motor u željenu baznu radnu tačku. Da bi se to postiglo, najpre se otvor leptira podesi na željenu vrednost, a onda se točkićem za podešavanje intenziteta kočenja na komandnom ormanu dinamometra (*Sl. 7.7*, na strani 174), moment kočenja dovodi na vrednost koja, prema *Tab. 7.8*, odgovara trenutnom otvoru leptira. Kada se broj obrtaja stabilizuje, treba proveriti da li je njegova vrednost približno jednaka odgovarajućoj vrednosti iz *Tab. 7.8*. Ova provera je izvršavana u svim odabranim baznim radnim tačkama, čime je potvrđena tačnost njihovog određivanja po proceduri, opisanoj u Odeljku 7.4.2.

Ovde treba istaći da je prilikom izvođenja svih eksperimenata, dinamometar bio podešen na režim sa konstantnim momentom kočenja. To znači da pri pojavi nekog poremećaja, kao što je neispravnost u usisnom sistemu motora, moment kočenja ostaje nepromenjen, a menja se broj obrtaja, ukoliko je poremećaj dovoljno veliki. Ovaj režim kočenja je izabran, jer je broj obrtaja motora parametar, koji je u svakom momentu dostupan pri vožnji automobila, pa se njegova promena može iskoristiti u dijagnostičke svrhe. Sa druge strane, na današnjim automobilima ne postoji davač obrtnog momenta motora, pa se promena te veličine ne može upotrebiti u dijagnostici.

Po uspostavljanju ustaljenog rada motora u željenoj baznoj radnoj tački, pomoću softvera *VemsTune* aktivira se opcija snimanja svih raspoloživih podataka u tzv. log-fajlu. U zavisnosti od broja raspoloživih senzora na motoru, broj promenljivih čiji se istorijat promene beleži u log-fajlu, može biti i preko 50. Međutim, za ovo istraživanje, direktno ili indirektno su interesantne sledeće veličine:

- broj obrtaja,
- apsolutni pritisak u usisnom kolektoru,
- otvor leptira,
- stvarna vrednost koeficijenta viška vazduha,
- željena vrednost koeficijenta viška vazduha,
- temperatura u usisnom kolektoru,
- temperatura rashladne tečnosti i
- dužina impulsa ubrizgavanja.

Dakle, kada se aktivira snimanje log-fajla, pristupa se izazivanju željene neispravnosti u usisnom sistemu. Ovde treba napomenuti da je, ako se radi o neispravnosti ulaza „falš“-vazduha u usisni sistem, čep sa odgovarajućim otvorom trebalo montirati na slavinu (*Sl. 7.15*, strana 180) još na početku eksperimenta, pre uspostavljanja željene bazne radne tačke motora. Po izazivanju željene neispravnosti (otvaranjem slavine ili promenom ofseta MAP senzora), dolazi do promene vrednosti nekih parametara motora. Broj obrtaja motora i koeficijent viška vazduha su najinertnije veličine, tj. ustaljivanje njihovih vrednosti posle poremećaja odvija se najsporije. Zato se njihove vrednosti prate i kada se ustale, ostavlja se motor da radi na tom ustaljenom režimu minimalno 6 sekundi, da bi se snimio dovoljan broj uzoraka za razvoj dijagnostičkog modela. Upravo taj period ustaljenog rada motora uz prisustvo neke neispravnosti je interesantan za istraživanje u ovom radu, a podaci snimljeni u tom vremenskom intervalu biće korišćeni za obuku i testiranje neuronske mreže. Po isteku tog vremena, motor se dovodi u ispravno stanje (zatvaranjem slavine ili postavljanjem ofseta MAP senzora na inicijalnu vrednost). Nakon minimalno pola minuta rada u ispravnom stanju, pristupa se izvođenju sledećeg eksperimenta, u novoj baznoj radnoj tački i/ili sa novim stanjem usisnog sistema.

## 7.5 Razvoj laboratorijskog dijagnostičkog modela usisnog sistema

Kako je ranije rečeno, ovaj model čini jedna neuronska mreže razvijena na osnovu podataka dobijenih u laboratorijskim eksperimentima. Razvoj pomenute neuronske mreže tekao je po istoj proceduri, koja je primenjena i pri razvoju neuronskih mreža iz simulacionog dijagnostičkog modela usisnog sistema. Zbog toga će iz prikaza načina na koji se došlo do laboratorijske neuronske mreže, koji sledi u nastavku, biti izostavljeni opširniji opisi pojedinih faza te procedure gde god je to moguće, a više pažnje će biti posvećeno eventualnim razlikama između laboratorijskog i simulacionog dijagnostičkog modela.

### 7.5.1 Definisane ulaznih i izlaznih veličina

S obzirom da je u pitanju motor, kod kojeg se maseni protok vazduha određuje računanjem, a ne direktnim merenjem, ulazne veličine su definisane po uzoru na simulacioni dijagnostički model za slučaj motora bez protokomera, ali su napravljene i neke izmene. U skladu sa tim, skup ulaznih veličina čine sledeće veličine:

- broj obrtaja motora,
- pritisak u usisnom kolektoru i
- otvor leptira.

Poređenjem sa skupom ulaznih veličina neuronskih mreža simulacionog dijagnostičkog modela, navedenim u *Tab. 6.3* na strani 127, uočava se da su u laboratorijskom modelu izostavljene temperatura okolnog vazduha i temperatura u usisnom kolektoru. Najvažniji doprinos ovih promenljivih u simulacionom dijagnostičkom modelu bio je u dijagnostikovanju neispravnosti EGR sistema. Pošto ispitivani motor nema EGR sistem, pomenute dve promenljive su, sa dijagnostičkog aspekta, izgubile na značaju, pa se smatralo da bi bilo racionalno da se one izostave iz laboratorijskog dijagnostičkog modela.

Princip definisanja izlaznih veličina neuronske mreže ostao je isti, kao i kod simulacionog dijagnostičkog modela. Broj izlaznih veličina je jednak broju razmatranih stanja usisnog sistema, tako da svaka izlazna veličina reprezentuje po jedno stanje. Imajući u vidu to, kao i sva razmatrana stanja, pobrojana u *Tab. 7.6*, skup izlaznih veličina činiće sledeći elementi: „SveOK“, „FaV2mm“, „FaV4mm“, „FaV6mm“, „MAP-10“, „MAP+10“ i „MAP+20“. Željena vrednost svake od ovih veličina je 1, ukoliko usisni sistem jeste u stanju

koje ta veličina predstavlja, a ako sistem nije u stanju koje ta veličina predstavlja, onda je željena vrednost te promenljive 0.

## 7.5.2 Prikupljanje i priprema podataka za obuku

Izvor podataka za obuku neuronske mreže bili su log-fajlovi, dobijeni pomoću softvera *VemsTune* snimanjem raspoloživih podataka tokom izvođenja svakog pojedinačnog merenja, kako je opisano u tački 7.4.3. Tamo je napomenuto da je pri svakom merenju motor radio minimalno 6 sekundi na ustaljenom režimu, pri nekom željenom stanju usisnog sistema. Upravo podaci snimljeni u tom periodu rada motora su interesantni sa stanovišta razvoja dijagnostičkog modela. Međutim, originalni log-fajlovi ne mogu se direktno koristiti u drugim softverima, pa su pomoću *VemsTune*-a konvertovani u Excel format. Pregledom tako dobijenih Excel-fajlova utvrđeno je da broj snimljenih uzoraka u jednoj sekundi iznosi 27, što bi značilo da broj uzoraka tokom pomenutih 6 sekundi iznosi 162. Kada se taj broj pomnoži sa 7 posmatranih stanja, pa onda sa 6 baznih radnih tačaka motora i na kraju sa dve različite temperature okoline, dobija se ukupno 13608 uzoraka u skupu za obuku neuronske mreže. To znači da bi ulazna matrica za obuku bila formata 13608x3, a matrica željenih izlaza formata 13608x7. Tokom preliminarnog istraživanja utvrđeno je da Matlab-ov alat za rad sa neuronskim mrežama ne može da sprovede obuku ni sa matricama znatno manjeg formata. Iz ovih razloga, odlučeno je da se značajno smanji broj uzoraka koji se odnose na jedno stanje usisnog sistema. Tako je broj od 162 uzorka, koji odgovara periodu od 6 sekundi, sveden na 68 uzoraka, koji odgovaraju periodu od 2.5 sekundi. Pošto se radi o ustaljenom radu motora, uzorci iz odstranjenih 3.5 sekundi ne sadrže podatke koji nose mnogo novih informacija za neuronsku mrežu. Zato je procenjeno da ovakva redukcija podataka neće značajno pogoršati performanse neuronske mreže. Treba napomenuti da su u svakom Excel-fajlu, zadržani uzorci iz proizvoljne dve i po sekunde (u okviru ustaljenog rada motora), a svi ostali uzorci su odstranjeni. Osim toga, u svakom Excel-fajlu zadržani su podaci o broju obrtaja motora, pritisku u usisnom kolektoru i otvoru leptira (jer su to ulazne veličine neuronske mreže), kao i podaci o temperaturi u usisnom kolektoru (iz informativnih razloga).

Tab. 7.9 Ilustracija redukovano skupa podataka snimljenih pri jednom merenju

No	$\alpha_{\text{PROC}}$ [%]	$n$ [ $\text{min}^{-1}$ ]	$p_i$ [kPa]	$t_i$ [°C]
1	16,8	2348	68,3	38
2	16,8	2336	67,8	38
3	16,8	2349	67,6	38
4	16,8	2336	68,7	38
5	16,8	2347	67,8	38
6	16,8	2335	68,3	38
7	16,8	2341	67,7	38
8	16,8	2349	68,5	38
.	.	.	.	.
.	.	.	.	.
.	.	.	.	.
61	16,8	2337	67,9	38
62	16,8	2334	68,1	38
63	16,8	2347	68,8	38
64	16,8	2351	68,7	38
65	16,8	2339	68,1	38
66	16,8	2347	68,7	38
67	16,8	2336	68,4	38
68	16,8	2340	68,8	38

Na opisan način, od početnog log-fajla dobija se Excel tabela formata 68x4, čiji je skraćeni oblik, samo radi ilustracije, dat u *Tab. 7.9*. Prva kolona i zaglavlje ove tabele ne postoje u izvornom Excel-fajlu, ovde su dodati samo zbog preglednosti. Jedna ovakva tabela sadrži ekstrahovane podatke dobijene u jednom eksperimentu, koji se odnosi na jedno stanje usisnog sistema u jednoj baznoj radnoj tački motora. Da bi se dobio skup podataka za obuku neuronske mreže, potrebno je bilo izvesti niz eksperimenata za sva posmatrana stanja usisnog sistema (*Tab. 7.6* na strani 179), u svim odabranim baznim radnim tačkama motora (*Tab. 7.8* na strani 184). Imajući u vidu rečeno, napravljen je plan izvođenja eksperimenata za prikupljanje podataka za obuku neuronske mreže, koji je prikazan u *Tab. 7.10*.

*Tab. 7.10 Plan izvođenja laboratorijskih eksperimenata – obuka neuronske mreže*

Broj eksp.	Bazna radna tačka motora	Stanje usisnog sistema	Broj eksp.	Bazna radna tačka motora	Stanje usisnog sistema
1	Otvor leptira 8%, Moment kočenja 26 Nm, Broj obrtaja 1742 min <sup>-1</sup>	SveOK	22	Otvor leptira 15%, Moment kočenja 37 Nm, Broj obrtaja 2273 min <sup>-1</sup>	SveOK
2		Fav2mm	23		Fav2mm
3		Fav4mm	24		Fav4mm
4		Fav6mm	25		Fav6mm
5		MAP-10	26		MAP-10
6		MAP+10	27		MAP+10
7		MAP+20	28		MAP+20
8	Otvor leptira 10%, Moment kočenja 30 Nm, Broj obrtaja 1948 min <sup>-1</sup>	SveOK	29	Otvor leptira 17%, Moment kočenja 40 Nm, Broj obrtaja 2392 min <sup>-1</sup>	SveOK
9		Fav2mm	30		Fav2mm
10		Fav4mm	31		Fav4mm
11		Fav6mm	32		Fav6mm
12		MAP-10	33		MAP-10
13		MAP+10	34		MAP+10
14		MAP+20	35		MAP+20
15	Otvor leptira 12%, Moment kočenja 34 Nm, Broj obrtaja 2160 min <sup>-1</sup>	SveOK	36	Otvor leptira 20%, Moment kočenja 45 Nm, Broj obrtaja 2597 min <sup>-1</sup>	SveOK
16		Fav2mm	37		Fav2mm
17		Fav4mm	38		Fav4mm
18		Fav6mm	39		Fav6mm
19		MAP-10	40		MAP-10
20		MAP+10	41		MAP+10
21		MAP+20	42		MAP+20

Ovaj plan eksperimenata sproveden je dva puta: jednom tokom zime, kada je temperatura u laboratoriji približno 15°C i jednom tokom leta, kada je temperatura u laboratoriji oko 30°C, kako bi se u neuronsku mrežu ugradio i eventualni uticaj ovog klimatskog faktora. To znači da su za potrebe prikupljanja podataka za obuku neuronske mreže izvedena 84 eksperimenata i kao posledica toga, dobijena su 84 log-fajla. Redukovanjem svakog od ovih fajlova, kako je malo ranije opisano, dobijena su 84 Excel-fajla, oblika prikazanog u *Tab. 7.9* (bez zaglavlja i prve kolone). Na osnovu ovih 84 Excel-fajlova može se generisati skup podataka za obuku neuronske mreže, pomoću ranije opisanog programa *generisiStrukturuZaObuku* (strana 131) koji je korišćen i tokom razvoja simulacionog dijagnostičkog modela. Međutim, da bi se mogli sagledati i iskontrolisati rezultati, koji su dobijeni tokom izvođenja laboratorijskih eksperimenata, bilo je neophodno grupisati ih tako, da sva stanja koja se odnose na jednu baznu radnu tačku budu u jednom fajlu (slično kao kod virtuelnih eksperimenata, u Poglavlju 5). Iz tog razloga, generisanje skupa podataka za obuku izvedeno je u dva koraka.

U prvom koraku, pomenuta 84 Excel-fajla grupisana su tako, da je dobijeno novih 12 Excel-fajlova. U svakom od tih 12 fajlova objedinjeni su rezultati merenja koji se odnose na po jednu baznu radnu tačku motora (6 baznih radnih tačkama u letnjim i 6 u zimskim

uslovima). Ovo grupisanje je urađeno pomoću namenski napisane računarske program-funkcije *grupisaneNeispravnosti*, koja i iscrtava grafičke prikaze rezultata merenja u svakoj baznoj radnoj tački motora, za sva razmatrana stanja usisnog sistema. Izvorni kod ove funkcije dat je u prilogu P5.2.

Drugi korak se realizuje pomoću programa *generisiStrukturuZaObuku*, koji generiše skup podataka za obuku neuronske mreže, koristeći pomenutih 12 Excel-fajlova, na potpuno isti način, kao kod simulacionog dijagnostičkog modela. Dobijeni skup podataka za obuku skladišti se u vidu jedne Matlab-strukture u odgovarajući Matlab-fajl, sa automatski generisanim imenom. Najbitnije komponente ove Matlab-strukture su *ulazna matrica za obuku* i *matrica željenih izlaza* neuronske mreže, koje su formata 5712x3 i 5712x7, respektivno. Broj 5712 označava ukupan broj uzoraka za obuku (84 merenja x 68 uzoraka po jednom merenju), 3 je broj ulaznih veličina, a 7 je broj izlaznih veličina. Oblik matrice željenih izlaza prikazan je na Sl. 7.18.

7 kolona

		SveOK	FaV2mm	FaV4mm	.	.	MAP+20	
Bazna radna tačka br. 1 (7 x 68 vrsta)	68 vrsta	1	0	0	.	.	0	
		1	0	0	.	.	0	
		...	...	...	...	...	...	
	68 vrsta	1	0	0	.	.	0	
		0	1	0	.	.	0	
		0	1	0	.	.	0	
	68 vrsta	0	0	1	.	.	0	
		0	0	1	.	.	0	
		0	0	1	.	.	0	
	68 vrsta	0	0	1	.	.	0	
		0	0	0	.	.	1	
		0	0	0	.	.	1	
	Bazna radna tačka br. 12 (7 x 68 vrsta)	68 vrsta	1	0	0	.	.	0
			1	0	0	.	.	0
			...	...	...	...	...	...
		68 vrsta	1	0	0	.	.	0
			0	1	0	.	.	0
			0	1	0	.	.	0
		68 vrsta	0	1	0	.	.	0
			0	0	1	.	.	0
			0	0	1	.	.	0
		68 vrsta	0	0	1	.	.	0
			0	0	0	.	.	1
			0	0	0	.	.	1
				0	0	.	.	1

5712 x 7

Sl. 7.18 Ilustracija matrice željenih izlaza neuronske mreže



Grafički prikazi rezultata merenja relevantnih veličina u svim baznim radnim tačkama motora i pri obe temperature okoline, koji su dobijeni pomoću programa *grupisaneNeispravnosti*, dati su na priloženom CD-ROM-u. Ovde će radi ilustracije, a i radi verifikacije ranije opisanog simulatora neispravnosti, biti data četiri dijagrama (Sl. 7.19 - Sl. 7.22), koji prikazuju rezultate merenja u dve bazne radne tačke, na obe temperature okoline. Treba istaći da je motor bio podešen da radi u režimu zatvorene sprege sa lambda-sondom, a zadržana je i fabrička podešenost VEMS-a na maksimalnoj korekciji sastava smeše do 20%, pri čemu je tokom preliminarnih merenja ustanovljeno da sistem nije uvek u mogućnosti da izvrši maksimalno dozvoljenu korekciju.

Poređenjem Sl. 7.19 sa Sl. 7.20, kao i Sl. 7.21 sa Sl. 7.22, može se uočiti da se one međusobno veoma malo razlikuju, jer su podaci, prikazani na slikama koje se porede, snimljeni u istoj baznoj radnoj tački, samo pri različitim temperaturama okoline.

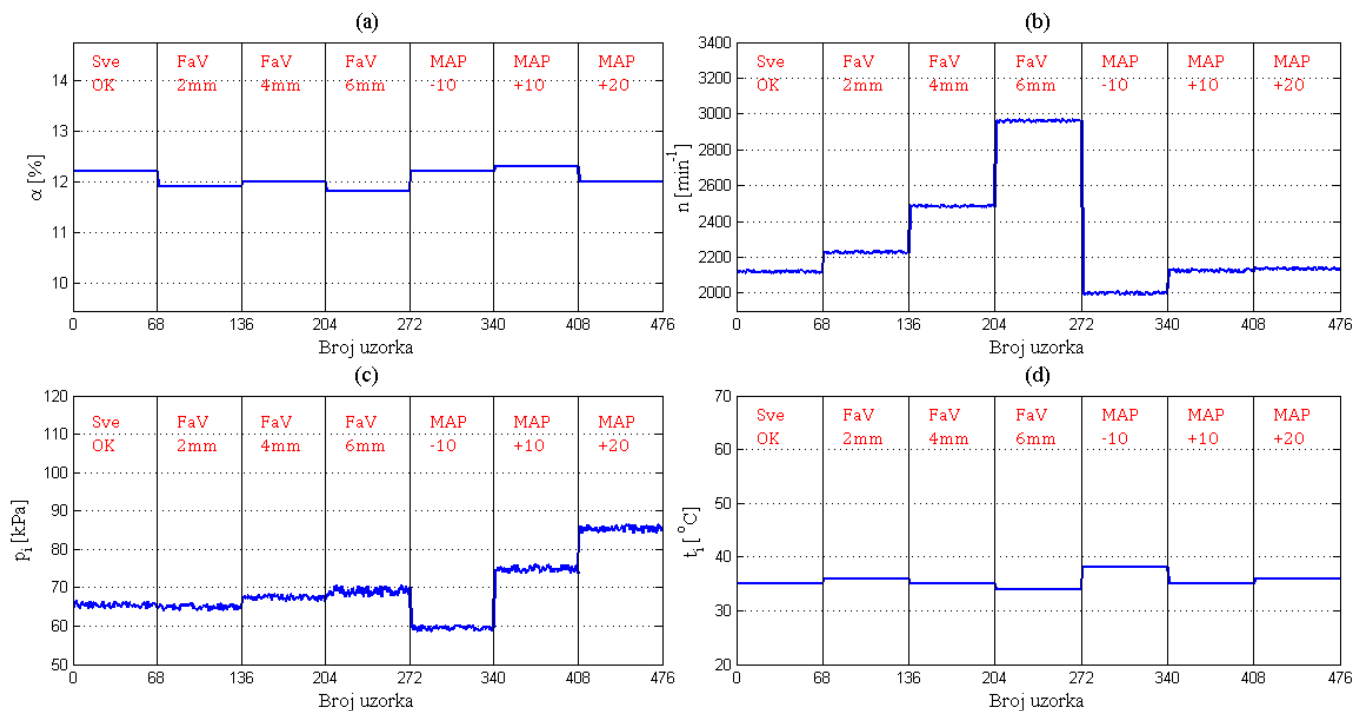
Uporednim posmatranjem istih stanja usisnog sistema na odgovarajućim dijagramima broja obrtaja (Sl. 7.19b - Sl. 7.20b, kao i Sl. 7.21b - Sl. 7.22b), zapaža se veoma blago uvećan broj obrtaja motora pri merenjima izvedenim u zimskim mesecima, u odnosu na letnje. Ovo je očekivano, jer je okolni vazduh bio hladniji i gušći, pa je motor razvijao nešto veću snagu. Pritisak u usisnom kolektoru, izmeren u zimskim mesecima, je nešto niži, što se uočava poređenjem odgovarajućih dijagrama pritiska u usisnom kolektoru (Sl. 7.19c - Sl. 7.20c, kao i Sl. 7.21c - Sl. 7.22c). Ovo važi, u većoj ili manjoj meri, za svih sedam stanja usisnog sistema. Ista pravilnost je uočena i u rezultatima dobijenim u virtuelnim eksperimentima.

Dalje, praćenjem uticaja pojedinih neispravnosti na broj obrtaja motora, uočava se da neispravnosti tipa „ulaz falš-vazduha u usisni sistem“ dovode do povećanja broja obrtaja motora i to utoliko većeg, što je intenzitet neispravnosti veći (Sl. 7.19b do Sl. 7.22b). Pri tome je uticaj ovog tipa neispravnosti veći pri manjim otvorima leptira.

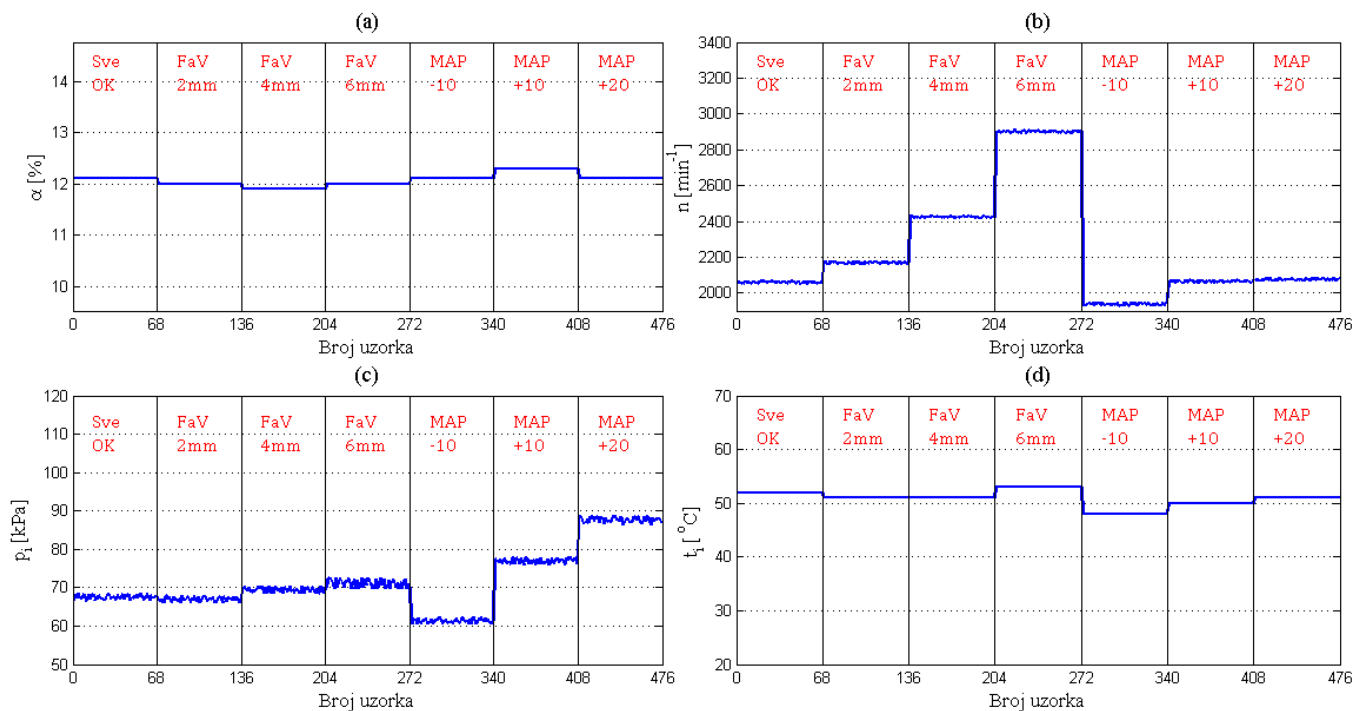
Što se tiče neispravnosti tipa „pogrešno očitavanje MAP senzora“, uticaj na broj obrtaja je znatno slabije izražen. Uočljivo je smanjenje broja obrtaja pri neispravnosti „MAP-10“, dok neispravnosti „MAP+10“ i „MAP+20“ gotovo da ne utiču na broj obrtaja (Sl. 7.19c do Sl. 7.22c). Ako se dijagram broja obrtaja sa bilo koje od četiri razmatrane slike uporedi sa dijagramom broja obrtaja na Sl. 5.34 (strana 113), može se primetiti da su uticaji pojedinih neispravnosti na broj obrtaja veoma slični (sa neznatnim odstupanjima). Inače, Sl. 5.34 je reper za poređenje, jer prikazuje rezultate virtuelnog eksperimenta za slučaj koji je najbližiji ispitivanom motoru (motor bez protokomera, koji radi u zatvorenoj sprezi sa lambda-sondom, pri čemu je korekcija vrednosti  $\lambda$  ograničena). Ovime rezultati, dobijeni pomoću simulatora neispravnosti, dobijaju još jednu eksperimentalnu potvrdu.

Kada je u pitanju uticaj pojedinih neispravnosti na pritisak u usisnom kolektoru, posmatranjem Sl. 7.19c do Sl. 7.22c, zapaža se da neispravnost „FaV2mm“ ima veoma mali uticaj, dok „FaV4mm“ i „FaV6mm“ dovode do приметnog porasta pritiska. Uticaj neispravnosti „MAP-10“, „MAP+10“ i „MAP+20“ na pritisak u usisnom kolektoru je približno takav, kako njihove oznake sugerišu. Poređenjem navedenih uticaja sa rezultatima odgovarajućeg virtuelnog eksperimenta, prikazanim na Sl. 5.35 (strana 113), dolazi se do zaključka da je i u ovom segmentu simulator neispravnosti dobio eksperimentalnu potvrdu.

Posmatranjem Sl. 7.19a do Sl. 7.22a može se uočiti da otvor leptira nije pri svim merenjima imao tačno deklarisanu vrednost, nego je bilo odstupanja reda veličine desetih delova procenta. Razlog tome je taj, što se na displeju *VemsTune*-a tokom merenja prikazuje zaokrugljena celobrojna vrednost otvora leptira (u procentima), a u log-fajlu se snima vrednost u tačnosti desetog dela procenta. Međutim, ovo ne samo da nema negativan efekat na skup podataka za obuku neuronske mreže, nego čak doprinosi da ovaj skup bude raznovrsniji, što može poboljšati performanse neuronske mreže.



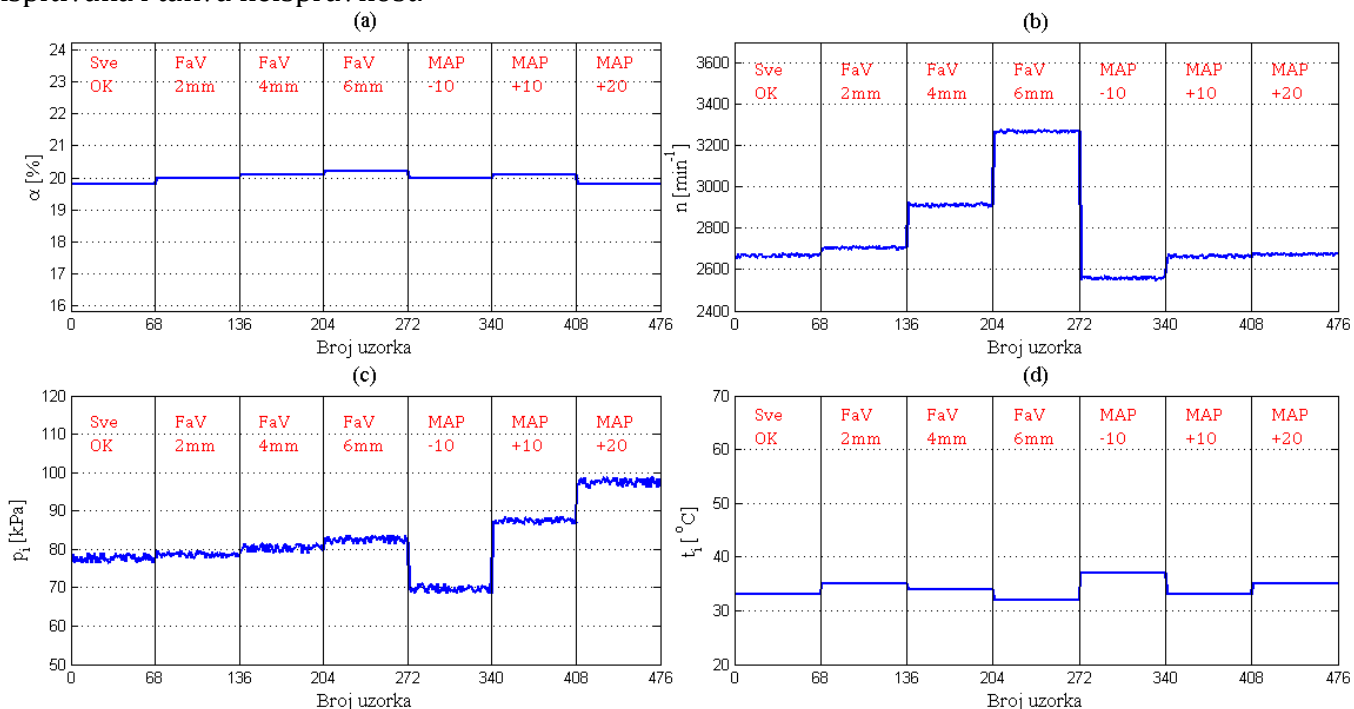
Sl. 7.19 Rezultati merenja u zimskim mesecima pri približnom otvoru leptira 12%, i momentu kočenja 34 Nm: (a) otvor leptira, (b) broj obrtaja motora, (c) apsolutni pritisak u usisnom kolektoru, (d) temperatura u usisnom kolektoru



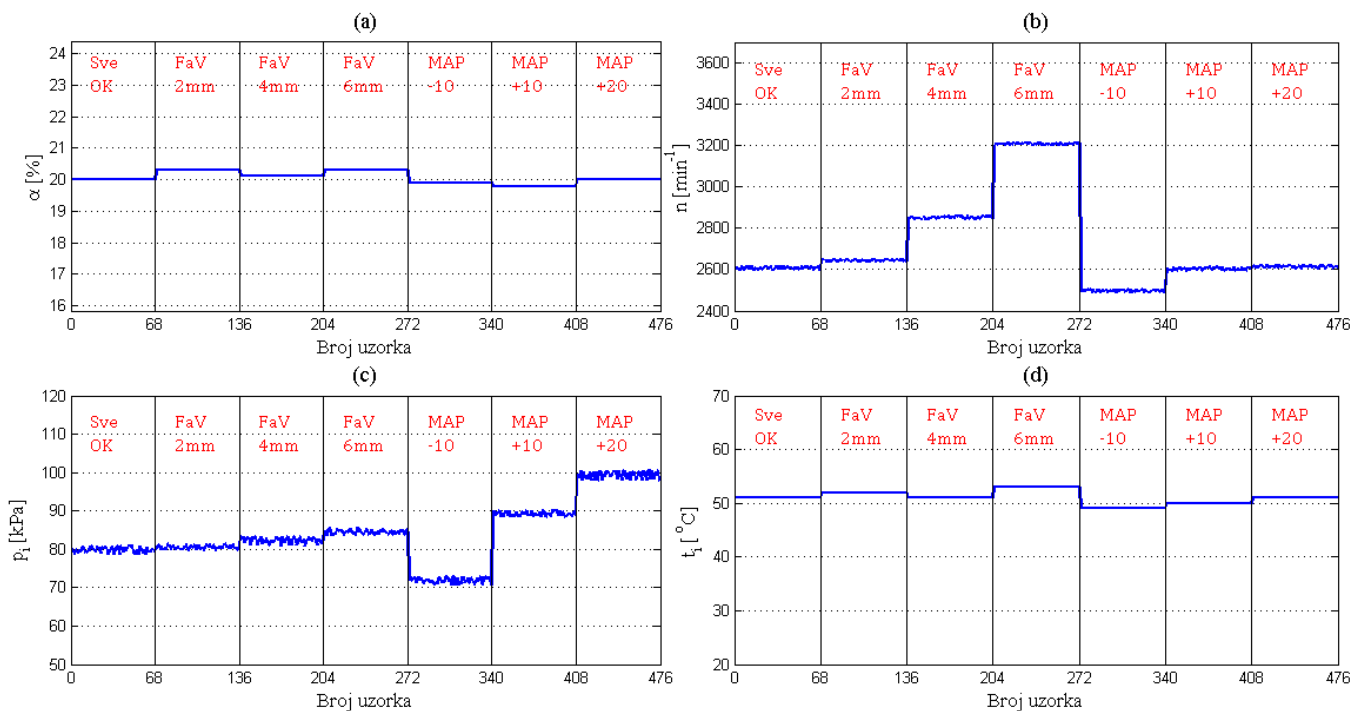
Sl. 7.20 Rezultati merenja u letnjim mesecima pri približnom otvoru leptira 12%, i momentu kočenja 34 Nm: (a) otvor leptira, (b) broj obrtaja motora, (c) apsolutni pritisak u usisnom kolektoru, (d) temperatura u usisnom kolektoru

Ranije je objašnjeno zašto temperatura u usisnom kolektoru nije svrstana u ulazne veličine neuronske mreže, a na Sl. 7.19d do Sl. 7.22d se to može i videti. Naime, jedina pravilnost koja se može uočiti na ovim slikama je ta, da su u letnjim mesecima izmerene više vrednosti temperature u usisnom kolektoru (oko 50°C), nego u zimskim (oko 35°C). Nikakva druga pravilnost, koja bi mogla pomoći u dijagnostikovanju neispravnosti, nije zapažena. To znači da pojava neke od posmatranih neispravnosti nema osetniji uticaj na temperaturu u usisnom kolektoru. Zbog toga je

ova veličina izostavljena iz skupa podataka za obuku neuronske mreže. Ovo nije bio slučaj kod simulacionog dijagnostičkog modela usisnog sistema, jer je model motora koji je korišćen u simulacijama imao EGR sistem, a praćene su i neispravnosti tipa „umanjen EGR protok“. Taj tip neispravnosti ima osetan uticaj na temperaturu u usisnom kolektoru, pa bi ova veličina svakako imala značajniju ulogu u obuci neuronske mreže, da je na konkretnom motoru mogla da bude ispitivana i takva neispravnost.



Sl. 7.21 Rezultati merenja u zimskim mesecima pri približnom otvoru leptira 20%, i momentu kočenja 45 Nm: (a) otvor leptira, (b) broj obrtaja motora, (c) apsolutni pritisak u usisnom kolektoru, (d) temperatura u usisnom kolektoru



Sl. 7.22 Rezultati merenja u letnjim mesecima pri približnom otvoru leptira 20%, i momentu kočenja 45 Nm: (a) otvor leptira, (b) broj obrtaja motora, (c) apsolutni pritisak u usisnom kolektoru, (d) temperatura u usisnom kolektoru

### 7.5.3 Strukturisanje i obuka neuronske mreže

S obzirom na zadovoljavajuće rezultate, koji su postignuti neuronskim mrežama u okviru simulacionog dijagnostičkog modela, nije bilo razloga da se u okviru laboratorijskog dijagnostičkog modela previše menjaju parametri strukture i obuke neuronske mreže. Jedina izmena koja je izvršena jeste izmena topologije neuronske mreže. Broj ulaznih i izlaznih neurona je smanjen zbog usklađivanja sa brojem ulaznih i izlaznih veličina, a broj neurona u skrivenim slojevima je umanjen, da bi se skratilo vreme trajanja obuke i da bi se dobilo na uštedi memorijskih resursa računara. Svi ostali parametri neuronskih mreža iz simulacionog dijagnostičkog modela su preslikani na ovu neuronsku mrežu. Zbog toga se ovde samo navode, bez obrazloženja, osnovne karakteristike ove mreže:

- To je BP neuronska mreža sa topologijom 3-15-20-7.
- Kao aktivacione funkcije u oba skrivena i u izlaznom sloju korišćene su bipolarne sigmoidne funkcije.
- Za obuku neuronske mreže korišćen je skalirani konjugovano-gradijentni BP algoritam.
- Funkcija greške je bila srednja kvadratna greška, sa ciljnom vrednošću 0.
- Koeficijent brzine učenja je bio 0.01.
- Uzajamni odnos veličina podskupova podataka za obuku, validaciju i testiranje bio je 60:20:20.
- Maksimalni broj ciklusa učenja je postavljen na 1000.
- Minimalna vrednost gradijenta greške je  $10^{-5}$ .
- Broj validacionih provera je 6

Praktična realizacija strukture neuronske mreže i njena obuka izvedeni su primenom programa *kreirajNetStrukturu*, koji je korišćen i u slučaju simulacionog dijagnostičkog modela, a opisan je u tački 6.3.6.

### 7.5.4 Testiranje neuronske mreže

Podaci za testiranje neuronske mreže dobijeni su izvođenjem eksperimentalnih merenja, na isti način, kao i podaci za obuku. Slično simulacionom dijagnostičkom modelu, i ovde je ideja bila da se testiranje neuronske mreže izvrši u nekoliko nivoa, koji su različiti težina sa aspekta dijagnostikovanja neispravnosti. U nastavku se daju detaljniji opisi ovih nivoa.

#### Nivo I

Da bi se izvršilo testiranje na ovom nivou, izvršeni su dodatni eksperimenti u istim onim baznim radnim tačkama, u kojima su snimani podaci za obuku neuronske mreže. Izvršeno je po jedno dodatno merenje pri svakom od sedam stanja usisnog sistema, u svakoj od posmatranih baznih radnih tačaka. Neka od ovih merenja vršena su u letnjim, neka u jesenjim, a neka u zimskim mesecima, a tom prilikom nije vođeno računa o temperaturi u laboratoriji, da bi testiranje bilo što objektivnije. Dodatni eksperimenti su izvedeni prema istom planu, koji je poslužio za prikupljanje podataka za obuku (*Tab. 7.10* na strani 187), tako da su izvedena ukupno 42 eksperimenta za potrebe testiranja na nivou I.

#### Nivo II

Zbog prikupljanja podataka za testiranje na nivou II, izvršen je određen broj eksperimenata u nepoznatim baznim radnim tačkama, u kojima su izazivane poznate neispravnosti. Među baznim radnim tačkama iz *Tab. 7.8* na strani 184, takve su dve: jedna se odnosi na otvor leptira 18%, a druga na otvor leptira 19%. Plan izvođenja eksperimenata za testiranje na ovom nivou dat je u *Tab. 7.11*.

Tab. 7.11 Plan izvođenja eksperimenata – testiranje neuronskih mreža (Nivo II)

Broj eksp.	Bazna radna tačka motora	Stanje usisnog sistema	Broj eksp.	Bazna radna tačka motora	Stanje usisnog sistema
1	Otvor leptira 18%, Moment kočenja 42 Nm, Broj obrtaja 2475 min <sup>-1</sup>	SveOK	8	Otvor leptira 19%, Moment kočenja 43 Nm, Broj obrtaja 2529 min <sup>-1</sup>	SveOK
2		Fav2mm	9		Fav2mm
3		Fav4mm	10		Fav4mm
4		Fav6mm	11		Fav6mm
5		MAP-10	12		MAP-10
6		MAP+10	13		MAP+10
7		MAP+20	14		MAP+20

Kako se vidi iz Tab. 7.11, u obe bazne radne tačke snimano je svih sedam poznatih stanja usisnog sistema, što čini ukupno 14 eksperimenata.

### Nivo III

Testiranje na nivou III zahtevalo je izvođenje eksperimenata u poznatim baznim radnim tačkama motora, ali sa izazivanjem neispravnosti koje su za neuronsku mrežu nepoznate. Tačnije, to su nepoznati intenziteti neispravnosti tipa „ulaz falš-vazduha u usisni sistem“ i „pogrešno očitavanje MAP senzora“. Ove neispravnosti date su u Tab. 7.12.

Tab. 7.12 Nepoznate neispravnosti izazivane u eksperimentima za testiranje

Naziv neispravnosti	Oznaka
MAP senzor očitava za 8 kPa nižu vrednost	MAP-8
MAP senzor očitava za 15 kPa višu vrednost	MAP+15
Ulaz „falš“-vazduha kroz otvor prečnika 3mm	FaV3mm
Ulaz „falš“-vazduha kroz otvor prečnika 5mm	FaV5mm

Eksperimenti, koji su izvedeni u cilju prikupljanja podataka za testiranje na nivou III, objedinjeni su u Tab. 7.13.

Tab. 7.13 Plan izvođenja eksperimenata – testiranje neuronskih mreža (Nivo III)

Broj eksp.	Bazna radna tačka motora	Stanje usisnog sistema	Broj eksp.	Bazna radna tačka motora	Stanje usisnog sistema
1	Otvor leptira 8%, Moment kočenja 26 Nm, Broj obrtaja 1742 min <sup>-1</sup>	Fav3mm	13	Otvor leptira 15%, Moment kočenja 37 Nm, Broj obrtaja 2273 min <sup>-1</sup>	Fav3mm
2		Fav5mm	14		Fav5mm
3		MAP-8	15		MAP-8
4		MAP+15	16		MAP+15
5	Otvor leptira 10%, Moment kočenja 30 Nm, Broj obrtaja 1948 min <sup>-1</sup>	Fav3mm	17	Otvor leptira 17%, Moment kočenja 40 Nm, Broj obrtaja 2392 min <sup>-1</sup>	Fav3mm
6		Fav5mm	18		Fav5mm
7		MAP-8	19		MAP-8
8		MAP+15	20		MAP+15
9	Otvor leptira 12%, Moment kočenja 34 Nm, Broj obrtaja 2160 min <sup>-1</sup>	Fav3mm	21	Otvor leptira 20%, Moment kočenja 45 Nm, Broj obrtaja 2597 min <sup>-1</sup>	Fav3mm
10		Fav5mm	22		Fav5mm
11		MAP-8	23		MAP-8
12		MAP+15	24		MAP+15

Kao što se vidi, u svakoj poznatoj baznoj radnoj tački izazivane su četiri nepoznate neispravnosti, pa broj izvedenih eksperimenata u ovoj fazi testiranja iznosi 24.

### Nivo IV

Ovaj nivo testiranja neuronske mreže je najopštiji, jer proverava sposobnost neuronske mreže da prepozna nepoznate neispravnosti u nepoznatim baznim radnim tačkama. Predstavlja neku kombinaciju nivoa II i nivoa III. Naime, pri izvođenju dodatnih eksperimenata izazivane su iste neispravnosti, kao pri testiranju na nivou III, u baznim radnim tačkama, koje su korišćene na nivou II. Tab. 7.14 prikazuje plan izvođenja eksperimenata na nivou IV, koji je obuhvatio 8 eksperimenata.

Tab. 7.14 Plan izvođenja eksperimenata – testiranje neuronskih mreža (Nivo IV)

Broj eksp.	Bazna radna tačka motora	Stanje usisnog sistema	Broj eksp.	Bazna radna tačka motora	Stanje usisnog sistema
1	Otvor leptira 18%, Moment kočenja 42 Nm, Broj obrtaja 2475 min <sup>-1</sup>	Fav3mm	5	Otvor leptira 19%, Moment kočenja 43 Nm, Broj obrtaja 2529 min <sup>-1</sup>	Fav3mm
2		Fav5mm	6		Fav5mm
3		MAP-8	7		MAP-8
4		MAP+15	8		MAP+15

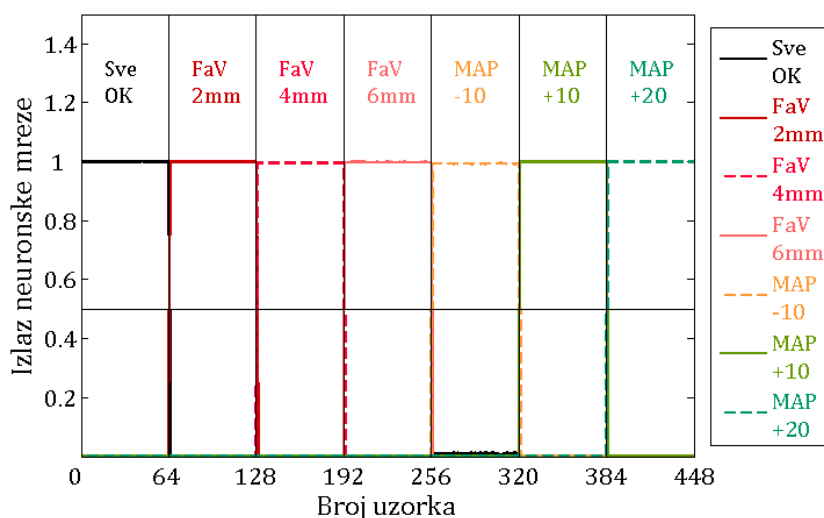
Praktična realizacija testiranja neuronske mreže na svim nivoima sprovedena je korišćenjem programa *testirajNetStrukturu* sa pratećim potprogramima, koji su primenjeni i u slučaju simulacionog dijagnostičkog modela, a opisani su u tački 6.3.6.

### 7.5.5 Rezultati testiranja neuronske mreže i diskusija

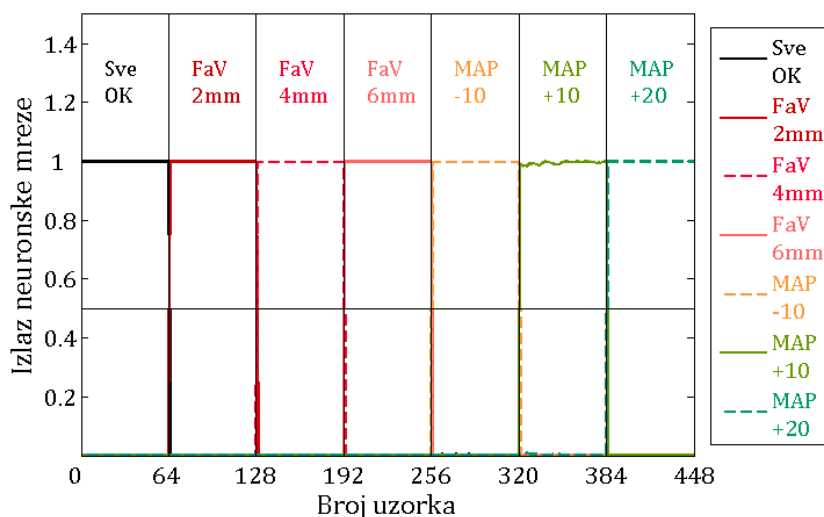
Kompletni rezultati testiranja neuronske mreže laboratorijskog dijagnostičkog modela, po pojedinim nivoima testiranja, daju se u ovom odeljku. Treba napomenuti da je i pri testiranju ove neuronske mreže, kao i kod simulacionog dijagnostičkog modela, primenjivana funkcija *preradiRedukujNNizlaz* (tačka 6.3.6), da bi se dobio odziv neuronske mreže, očišćen od pojedinačnih pikova, koji pogoršavaju njene performanse. Pri tome je prerađeni izlaz neuronske mreže računat kao srednja vrednost njenog trenutnog neprerađenog izlaza i četiri prethodna neprerađena izlaza. Zbog toga je broj uzoraka, koji odgovaraju jednom stanju usisnog sistema, sveden na 64, umesto 68, koliko ih je bilo u skupu za obuku. U nastavku se, kao konačni rezultati testiranja, prikazuju dijagrami koji predstavljaju prerađene izlaze neuronske mreže. Logika vizuelnih prikaza rezultata testiranja je ista, kao kod simulacionog modela, ali postoji jedna mala razlika. Naime, uz svaki dijagram je prikazana i legenda, pomoću koje se vidi, koja linija predstavlja odziv kojeg izlaznog neurona. Oznake u gornjem delu svakog dijagrama zadržale su isto značenje, kao kod simulacionog modela – za svaki uzorak testiranja pokazuju pri kojem stanju usisnog sistema je snimljen.

#### Nivo I – poznata stanja u poznatim baznim radnim tačkama

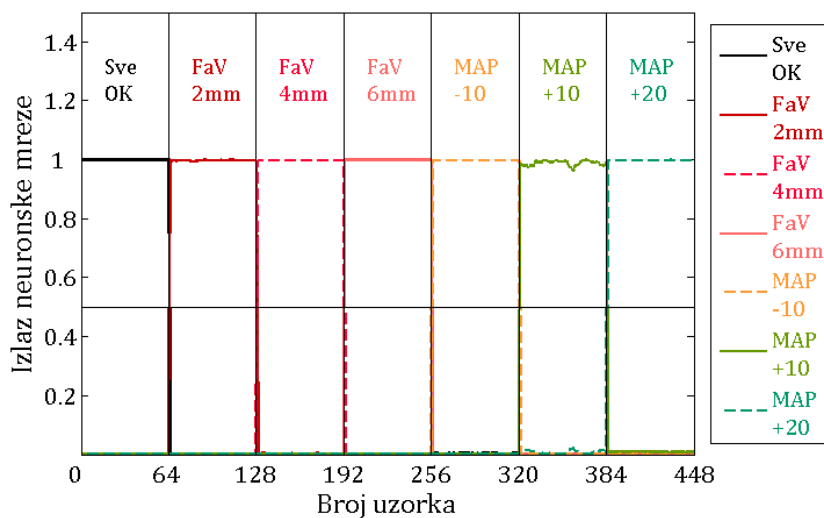
Rezultati testiranja na prvom nivou su, prema očekivanjima, izuzetno dobri, jer nema nepoznanica za neuronsku mrežu. Kako pokazuju Sl. 7.23 - Sl. 7.28, u svih 6 baznih radnih tačaka, neuronska mreža je prepoznala sva stanja usisnog sistema, na svim prezentovanim uzorcima za testiranje. I ne samo da ih je prepoznala, nego su odzivi izlaznih neurona u većini slučajeva jednaki jedinici ili vrlo bliski toj vrednosti. Jedino na Sl. 7.27 i Sl. 7.28, u slučaju stanja „FaV2mm“ odziv odgovarajućeg izlaznog neurona malo odstupa od jedinice, ali je osetno iznad linije praga identifikacije stanja. Ovakav rezultat je logičan, jer je prirodno da pri malo većim otvorima leptira (koje prikazuju Sl. 7.27 i Sl. 7.28), strujanje „falš“-vazduha kroz otvor prečnika 2mm, bude teže dijagnostikovano.



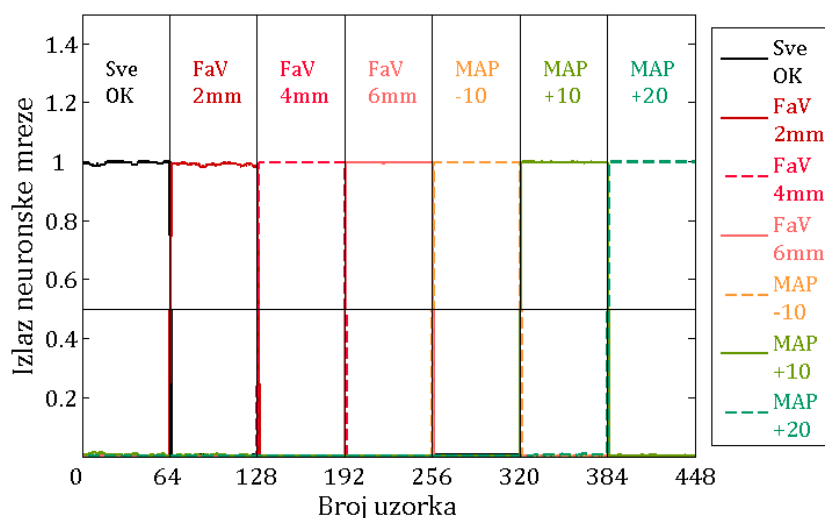
Sl. 7.23 Rezultati testiranja poznatim stanjima u poznatoj baznoj radnoj tački: otvor leptira  $\approx 8\%$ , moment kočenja  $\approx 26\text{Nm}$ , broj obrtaja  $\approx 1742\text{ min}^{-1}$



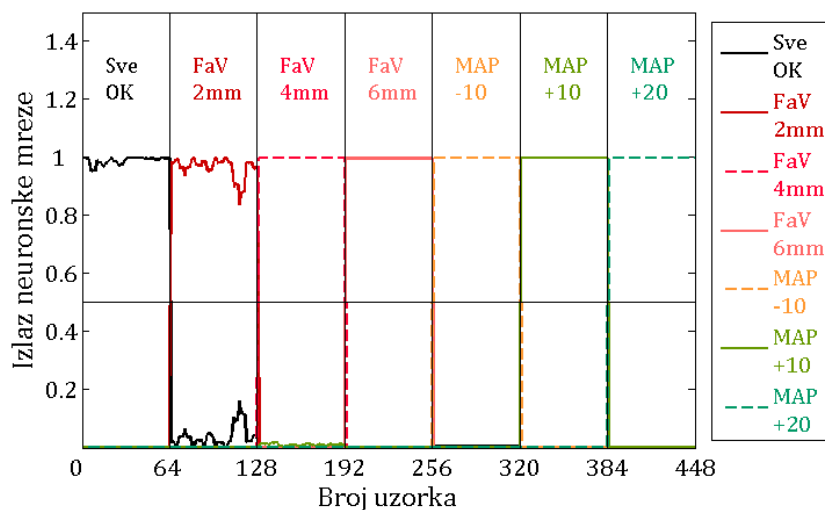
Sl. 7.24 Rezultati testiranja poznatim stanjima u poznatoj baznoj radnoj tački: otvor leptira  $\approx 10\%$ , moment kočenja  $\approx 30\text{Nm}$ , broj obrtaja  $\approx 1948\text{ min}^{-1}$



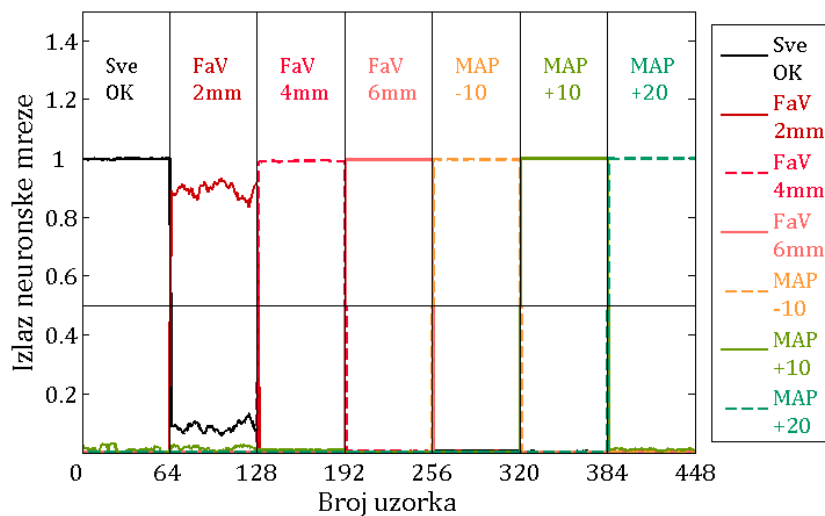
Sl. 7.25 Rezultati testiranja poznatim stanjima u poznatoj baznoj radnoj tački: otvor leptira  $\approx 12\%$ , moment kočenja  $\approx 34\text{Nm}$ , broj obrtaja  $\approx 2160\text{ min}^{-1}$



Sl. 7.26 Rezultati testiranja poznatim stanjima u poznatoj baznoj radnoj tački: otvor leptira  $\approx 15\%$ , moment kočenja  $\approx 37\text{Nm}$ , broj obrtaja  $\approx 2273\text{ min}^{-1}$



Sl. 7.27 Rezultati testiranja poznatim stanjima u poznatoj baznoj radnoj tački: otvor leptira  $\approx 17\%$ , moment kočenja  $\approx 40\text{Nm}$ , broj obrtaja  $\approx 2392\text{ min}^{-1}$

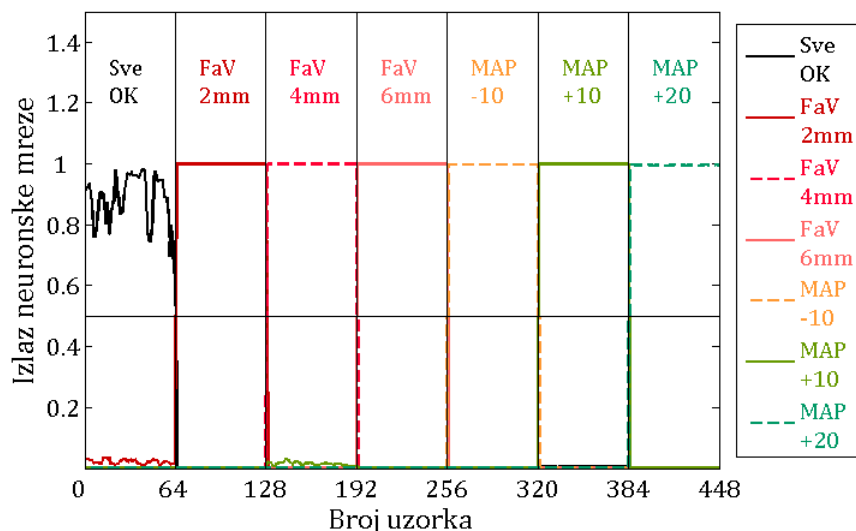


Sl. 7.28 Rezultati testiranja poznatim stanjima u poznatoj baznoj radnoj tački: otvor leptira  $\approx 20\%$ , moment kočenja  $\approx 45\text{Nm}$ , broj obrtaja  $\approx 2597\text{ min}^{-1}$

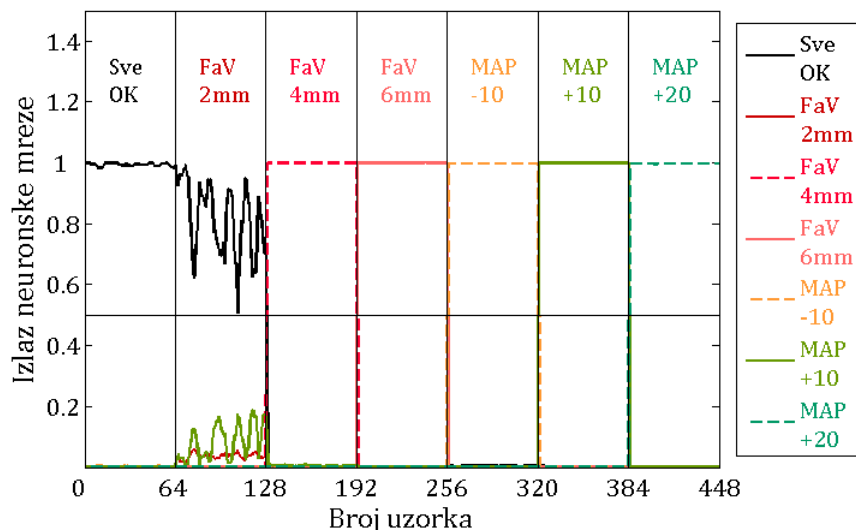


## Nivo II - poznata stanja u nepoznatim baznim radnim tačkama

Rezultati testiranja neuronske mreže na nivou II prikazani su na *Sl. 7.29* i *Sl. 7.30*, na kojima se može videti da je neuronska mreža prepoznala sva stanja na svim uzorcima, sa jednim izuzetkom. Naime, na pojavu strujanja „falš“-vazduha kroz otvor prečnika 2mm, pri otvoru leptira 19%, neuronska mreža reaguje kao da je sve u redu, što se vidi na *Sl. 7.30*. Objašnjenje ovakve reakcije neuronske mreže na ovu neispravnost dato je već u tumačenju rezultata testiranja na nivou I, jer je tamo primećena slična pojava, ali sa blažim posledicama.



*Sl. 7.29* Rezultati testiranja poznatim stanjima u nepoznatoj baznoj radnoj tački: otvor leptira  $\approx 18\%$ , moment kočenja  $\approx 42\text{Nm}$ , broj obrtaja  $\approx 2475\text{ min}^{-1}$

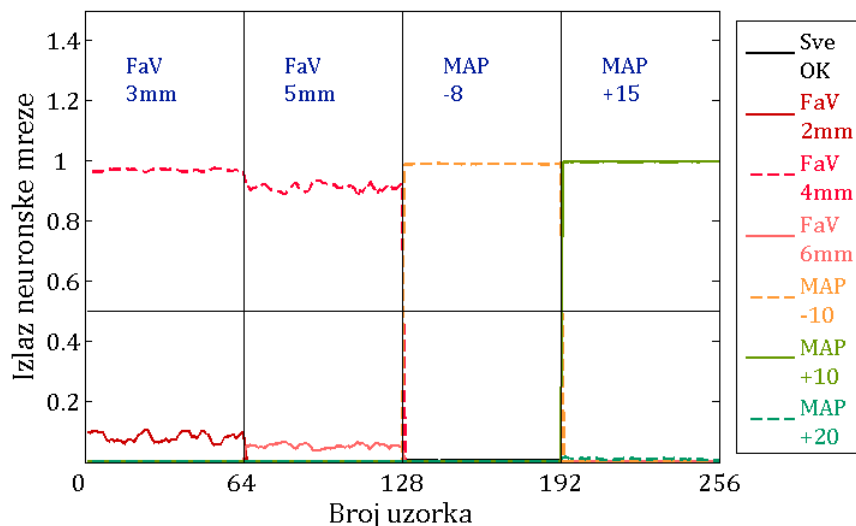


*Sl. 7.30* Rezultati testiranja poznatim stanjima u nepoznatoj baznoj radnoj tački: otvor leptira  $\approx 19\%$ , moment kočenja  $\approx 43\text{Nm}$ , broj obrtaja  $\approx 2529\text{ min}^{-1}$

## Nivo III - nepoznata stanja u poznatim baznim radnim tačkama

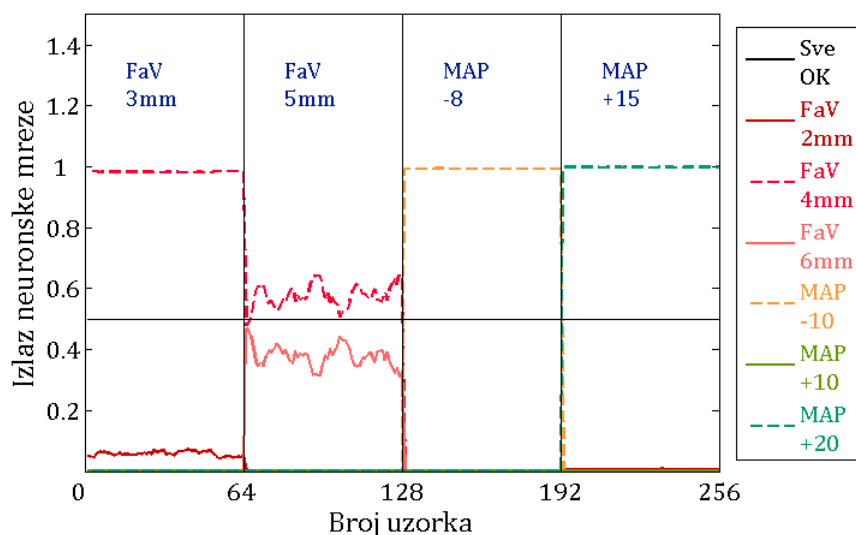
Rezultati testiranja na nivou III, sa četiri neispravnosti koje nisu korišćene prilikom snimanja podataka za obuku neuronske mreže, prikazani su na *Sl. 7.31* - *Sl. 7.36* za šest poznatih radnih tačkama, u kojima su vršena merenja. Globalni zaključak bi ponovo mogao da glasi da su gotovo sve neispravnosti u svim baznim radnim tačkama uspešno prepoznate, a u nastavku slede zapažanja u vezi sa svakom pojedinačnom baznom radnom tačkom.

Na Sl. 7.31 vidi se da neuronska mreža, na neispravnosti „FaV3mm“ i „FaV5mm“ reaguje kao da je u pitanju neispravnost „FaV4mm“, što se može smatrati uspešnim dijagnostikovanjem, jer je pogođen tip, a približno i intenzitet neispravnosti. Ovo važi za sve uzorke koji su korišćeni pri testiranju. Slična konstatacija se može izneti i po pitanju neispravnosti „MAP-8“ i „MAP+15“, na koje neuronska mreža reaguje kao da su u pitanju „MAP-10“ i „MAP+10“, respektivno, što se takođe smatra uspešnim dijagnostikovanjem.



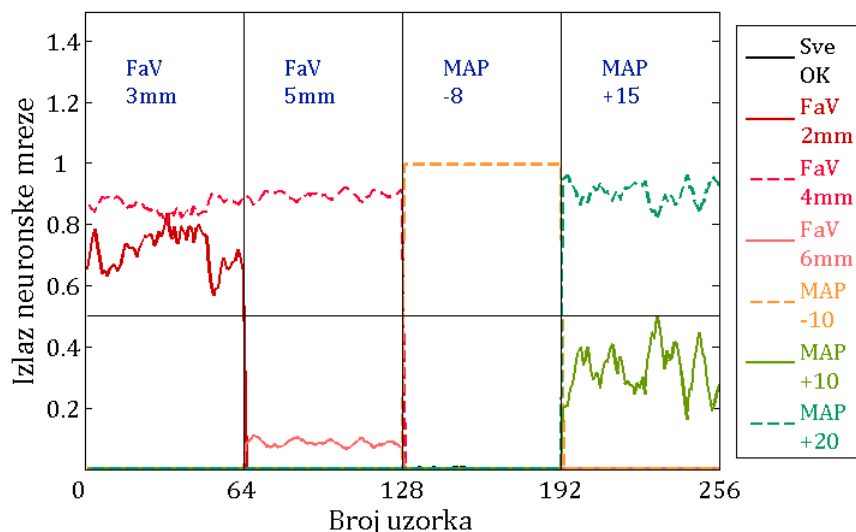
Sl. 7.31 Rezultati testiranja nepoznatim stanjima u poznatoj baznoj radnoj tački: otvor leptira  $\approx 8\%$ , moment kočenja  $\approx 26\text{Nm}$ , broj obrtaja  $\approx 1742\text{ min}^{-1}$

Sa aspekta dijagnostikovanja nepoznatih neispravnosti, Sl. 7.32 pokazuje gotovo isti rezultat kao prethodna slika, jedino što je kod neispravnosti „FaV5mm“ odziv odgovarajućeg izlaznog neurona relativno blizu linije praga prepoznavanja.



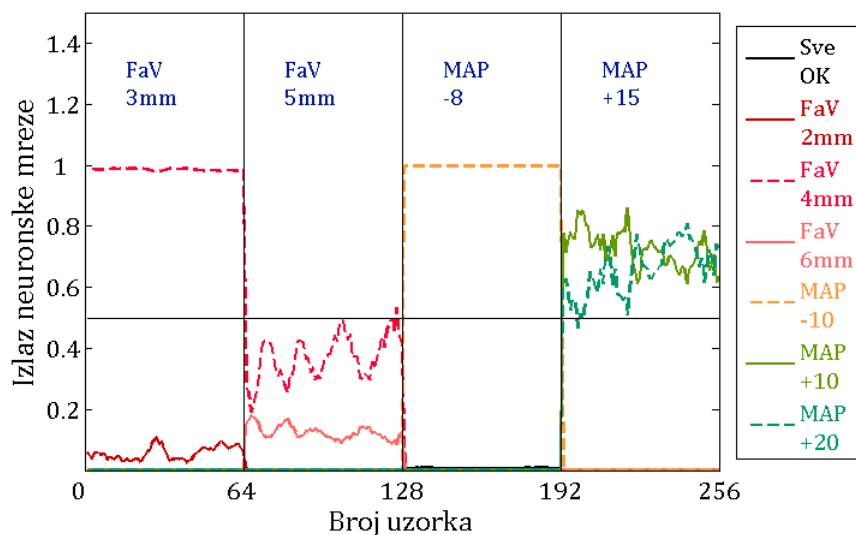
Sl. 7.32 Rezultati testiranja nepoznatim stanjima u poznatoj baznoj radnoj tački: otvor leptira  $\approx 10\%$ , moment kočenja  $\approx 30\text{Nm}$ , broj obrtaja  $\approx 1948\text{ min}^{-1}$

Nešto drugačija situacija je na Sl. 7.33, ali je uspešnost dijagnostikovanja neispravnosti i dalje stopostotna. Na neispravnost „FaV3mm“, alarmiraju dva izlazna neurona: „FaV2mm“ i „FaV4mm“, što je veoma dobra reakcija neuronske mreže. Dalje, neuronska mreža „vidi“ neispravnost „FaV5mm“ kao „FaV4mm“, neispravnost „MAP-8“ kao „MAP-10“ i neispravnost „MAP+15“ kao „MAP+20“, što su sve zadovoljavajući dijagnostički rezultati.



Sl. 7.33 Rezultati testiranja nepoznatim stanjima u poznatoj baznoj radnoj tački: otvor leptira  $\approx 12\%$ , moment kočenja  $\approx 34\text{Nm}$ , broj obrtaja  $\approx 2160\text{ min}^{-1}$

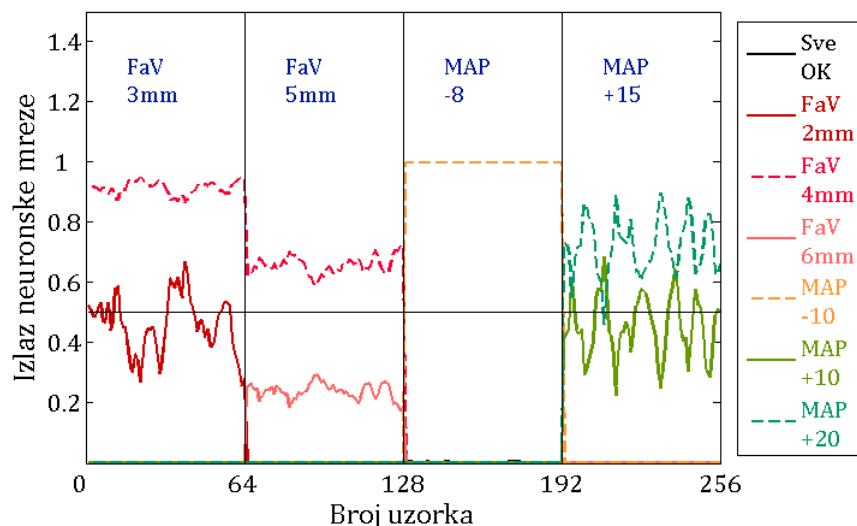
Sl. 7.34 pokazuje jednu neispravnost, koja nije uspešno dijagnostikovana. To je neispravnost „FaV5mm“, jer samo na uzorku broj 126 alarmira izlazni „FaV4mm“ neuron, a na svim ostalim uzorcima ni jedan izlazni neuron nema odziv iznad linije praga. Međutim, pošto ni izlazni „SveOK“ neuron ne signalizira da je sve u redu, to znači da je neuronska mreže detektovala neku neispravnost, ali je nije prepoznala. Neispravnosti „FaV3mm“ i „MAP-8“ su uspešno dijagnostikovane na isti način, kao u prve dve analizirane bazne radne tačke (Sl. 7.31 i Sl. 7.32). U slučaju neispravnosti „MAP+15“, alarmirala su dva izlazna neurona: „MAP+10“ i „MAP+20“, što se može smatrati ispravnom reakcijom neuronske mreže.



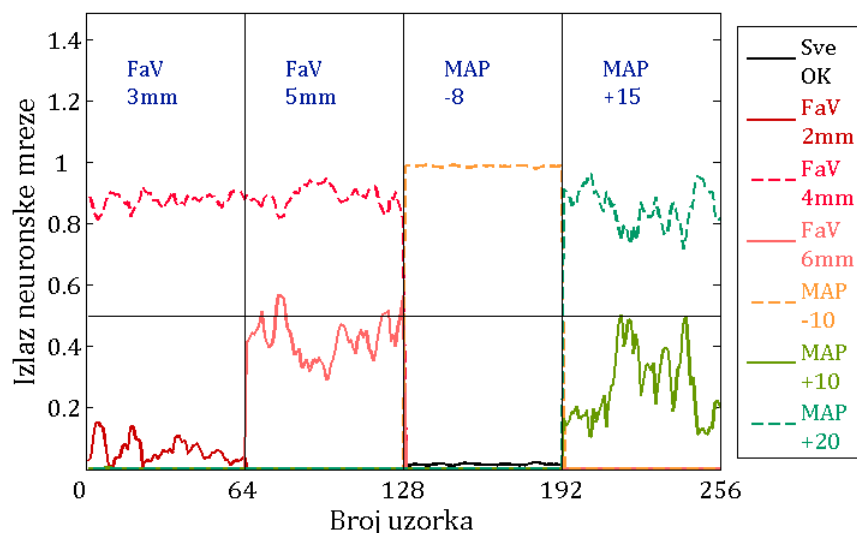
Sl. 7.34 Rezultati testiranja nepoznatim stanjima u poznatoj baznoj radnoj tački: otvor leptira  $\approx 15\%$ , moment kočenja  $\approx 37\text{Nm}$ , broj obrtaja  $\approx 2273\text{ min}^{-1}$

Rezultati dijagnostikovanja neispravnosti, prikazani na Sl. 7.35, su veoma dobri. Sve neispravnosti su na svim uzorcima uspešno prepoznate, s tim da se u slučaju neispravnosti „FaV3mm“ i „MAP+15“, povremeno pojavljuju dvostruki alarmi, ali su u pitanju alarmi odgovarajućih neurona.

Sl. 7.36 takođe pokazuje primer uspešnog dijagnostikovanja na svim uzorcima, sa nekoliko dvostrukih alarma kod neispravnosti „FaV5mm“. Kao u prethodnom slučaju, i ovde su u pitanju alarmi odgovarajućih izlaznih neurona.



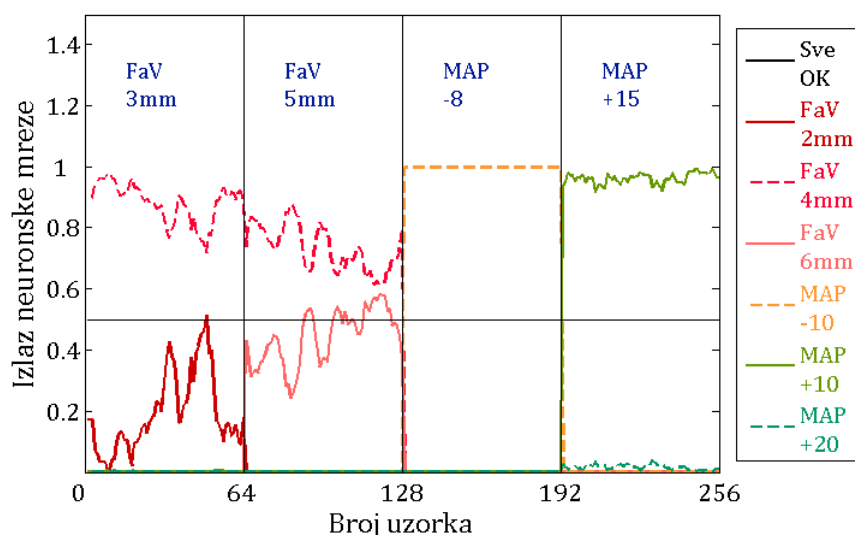
Sl. 7.35 Rezultati testiranja nepoznatim stanjima u poznatoj baznoj radnoj tački: otvor leptira  $\approx 17\%$ , moment kočenja  $\approx 40\text{Nm}$ , broj obrtaja  $\approx 2392\text{ min}^{-1}$



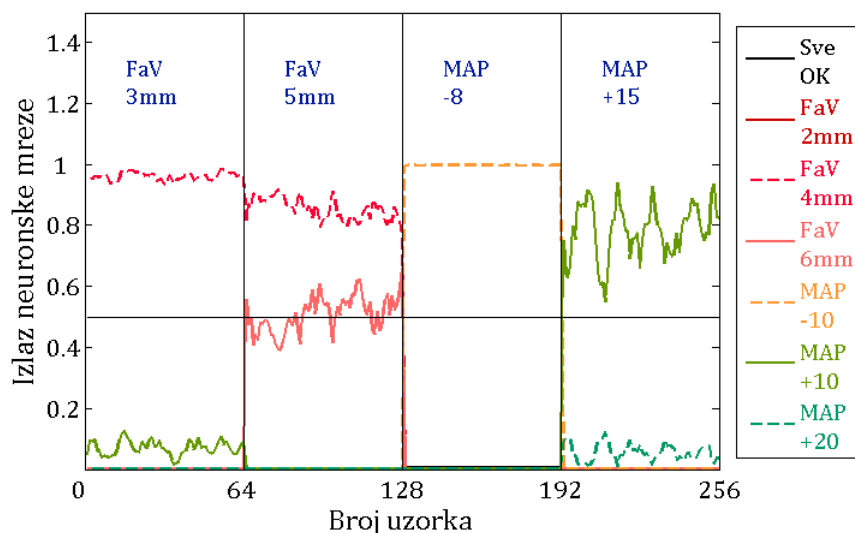
Sl. 7.36 Rezultati testiranja nepoznatim stanjima u poznatoj baznoj radnoj tački: otvor leptira  $\approx 20\%$ , moment kočenja  $\approx 45\text{Nm}$ , broj obrtaja  $\approx 2597\text{ min}^{-1}$

#### Nivo IV - nepoznata stanja u nepoznatim baznim radnim tačkama

Na ovom nivou, neuronska mreža je testirana u dve bazne radne tačke, za slučajeve iste četiri neispravnosti, kao na nivou III. Dobijeni rezultati su prikazani na Sl. 7.37 i Sl. 7.38. Kako pokazuju ove slike, u obe bazne radne tačke sve četiri neispravnosti su u potpunosti uspešno dijagnostikovane na svim uzorcima koji su predstavljeni neuronskoj mreži. Pri neispravnostima „FaV3mm“ i „FaV5mm“ alarmira uglavnom izlazni „FaV4mm“ neuron. Na određenom broju uzoraka, pri neispravnosti „FaV5mm“ pojavljuje se i alarm izlaznog „FaV6mm“ neurona, što je zadovoljavajuća reakcija neuronske mreže. Neispravnosti „MAP-8“ i „MAP+15“ neuronska mreža „vidi“ kao „MAP-10“ i „MAP+10“, respektivno, što se takođe može smatrati zadovoljavajućim rezultatom.



Sl. 7.37 Rezultati testiranja nepoznatim stanjima u nepoznatoj baznoj radnoj tački: otvor leptira  $\approx 18\%$ , moment kočenja  $\approx 42\text{Nm}$ , broj obrtaja  $\approx 2475\text{ min}^{-1}$



Sl. 7.38 Rezultati testiranja nepoznatim stanjima u nepoznatoj baznoj radnoj tački: otvor leptira  $\approx 19\%$ , moment kočenja  $\approx 43\text{Nm}$ , broj obrtaja  $\approx 2529\text{ min}^{-1}$

### 7.5.6 Zaključni komentar eksperimentalnih rezultata

Rezultati, dobijeni izazivanjem neispravnosti na stvarnom motoru koji radi u laboratorijskim uslovima, potvrdili su da simulator neispravnosti, razvijen i realizovan u ranijoj fazi istraživanja, prilično verno odslikava promene relevantnih motorskih veličina usled pojave konkretnih neispravnosti. Naime, ispitivanjem motora bez protokomera vazduha pri radu sa neispravnostima tipa „ulaz falš-vazduha u usisni sistem“ i „pogrešno očitavanje MAP senzora“, utvrđeno je da se broj obrtaja motora i pritisak u usisnom kolektoru menjaju upravo onako, kako je predviđeno simulatorom neispravnosti za tu vrstu motora i pomenute tipove neispravnosti.

Time je izvršena eksperimentalna verifikacija jednog segmenta simulatora neispravnosti. Da bi ova verifikacija bila potpuna, neophodno je izvršiti dodatna ispitivanja sa neispravnostima tipa „pogrešno očitavanje senzora temperature u usisnom kolektoru“ i „umanjen EGR protok“ na motoru bez protokomera, kao i ispitivanje motora sa masenim protokomera vazduha pri radu sa sva četiri tipa neispravnosti. Zbog ograničenih

laboratorijskih resursa pomenuta ispitivanja su izostala u okviru ovog istraživanja, ali bi mogla da nađu svoje mesto u nekom budućem istraživanju.

Slično simulatoru neispravnosti, i testiranjem razvijenog laboratorijskog dijagnostičkog modela, dobijeni su veoma dobri rezultati na svim nivoima.

Kada su u pitanju poznate neispravnosti, koje se pojavljuju na poznatim režimima rada, dijagnostikovanje se pokazalo 100-procentno uspešnim, jer su prepoznate sve neispravnosti na svim testiranim uzorcima. Nešto lošije su bile performanse dijagnostičkog modela pri prepoznavanju poznatih neispravnosti na nepoznatim režimima rada. Na nekima od tih režima dijagnostički model nije detektovao ulazak „falš“-vazduha kroz otvor prečnika 2 mm, ali su sve ostale neispravnosti u potpunosti dijagnostikovane na svim uzorcima.

Pri radu motora na poznatim režimima, ali sa nepoznatim neispravnostima, laboratorijski dijagnostički model je uspešno prepoznao gotovo sve tipove neispravnosti i vrlo približno odredio njihovu veličinu. Neuspešnim se pokazao samo u slučaju neispravnosti „ulaz falš-vazduha kroz otvor prečnika 5 mm“ i to samo na jednom testiranom režimu rada. Međutim, i u tom slučaju, detektovano je da postoji neka neispravnost, ali ona nije prepoznata. Ako bi prag identifikacije bio niži (npr. 0.3 umesto 0.5), uspešnost dijagnostike bi i u ovom slučaju bila stoprocentna.

Konačno, pri testiranju dijagnostičkog modela nepoznatim neispravnostima na nepoznatim režimima rada, uspešno su prepoznati svi tipovi neispravnosti, a njihova veličina je vrlo približno određena na svim testiranim uzorcima.

## 8 ZAKLJUČAK I SMERNICE DALJEG ISTRAŽIVANJA

U okviru istraživanja, opisanog u ovom radu, u potpunosti su realizovani ciljevi koji su postavljeni na njegovom početku.

Jedan od dva glavna cilja bio je da se prouči ponašanje automobilskog oto motora pri radu na ustaljenom režimu, kada u usisnom sistemu tog motora postoji jedna od sledećih neispravnosti: „nepredviđeni ulazak vazduha u usisni sistem“, „pogrešno očitavanje senzora pritiska u usisnom kolektoru“, „pogrešno očitavanje senzora temperature u usisnom kolektoru“ i „umanjeni EGR protok“. Istraživanje ponašanja motora u pomenutim situacijama sprovedeno je najpre simuliranjem pobrojanih neispravnosti pomoću namenski napisanog softvera koji je nazvan simulatorom neispravnosti. U simulatoru neispravnosti implementiran je jedan matematički model motora, koji je autor ovog rada razvio koristeći Hendriksov osrednjeni model. Tako dobijeni simulator neispravnosti predstavlja originalni naučni doprinos ove disertacije, s obzirom da uvidom u veliki broj publikacija iz ove oblasti nije pronađeno slično rešenje. Simulator neispravnosti omogućava da se prate relevantne motorske veličine, kao što su npr. broj obrtaja motora, pritisak i temperatura u usisnom kolektoru, maseni protok vazduha kroz usisni kolektor i vrednost koeficijenta viška vazduha, pri postojanju ranije pobrojanih neispravnosti u usisnom sistemu motora. Još jedna originalna karakteristika razvijenog simulatora neispravnosti je mogućnost da se simulira uticaj ranije navedenih neispravnosti na relevantne motorske veličine i kod motora sa direktnim (putem merenja) i kod motora sa indirektnim (putem računanja) određivanjem masenog protoka vazduha. Osim toga, pri bilo kojem od ova dva koncepta određivanja masenog protoka vazduha, motor može raditi u zatvorenoj ili u otvorenoj sprezi sa lambda-sondom. Oba pomenuta lambda-režima pokrivena su simulatorom neispravnosti, što je još jedan njegov pomak napred u odnosu na dosadašnja istraživanja iz ove oblasti. Dalje, u okviru simulatora neispravnosti integrisane su opšta jednačina uzdužne dinamike vozila i prateće jednačine za određivanje otpora kretanja vozila, čime je omogućeno simuliranje uticaja uzdužnog nagiba puta, stepena prenosa u kojem se vozilo kreće, brzine vetra, mase vozila itd. na relevantne motorske veličine. Pomenuti segment simulatora neispravnosti takođe predstavlja originalno rešenje u poređenju sa objavljenim radovima iz ove oblasti.

U okviru ove disertacije izvršeno je i ispitivanje ponašanja stvarnog automobilskog motora u laboratorijskim uslovima, pri postojanju neispravnosti tipa „nepredviđeni ulazak vazduha u usisni sistem“ i „pogrešno očitavanje senzora pritiska u usisnom kolektoru“. U pitanju je bio motor sa indirektnim određivanjem masenog protoka vazduha (bez protokomera), koji je prilikom snimanja eksperimentalnih podataka radio u režimu zatvorene sprege sa lambda-sondom. Posmatrano sa aspekta uticaja pomenutih neispravnosti na relevantne motorske veličine, pokazalo se da rezultati, dobijeni merenjima na stvarnom motoru, imaju sasvim zadovoljavajući nivo slaganja sa rezultatima, dobijenim pomoću simulatora neispravnosti za motor bez protokomera vazduha koji radi u zatvorenoj sprezi sa lambda-sondom.

U okviru laboratorijskog dela istraživanja nije izvršeno proučavanje ponašanja stvarnog motora za slučajeve neispravnosti tipa „pogrešno očitavanje senzora temperature u usisnom kolektoru“ i „umanjen EGR protok“. Neispravnost tipa „umanjen EGR protok“ nije ni mogla da bude razmatrana na stvarnom motoru, jer taj motor nije ni imao EGR sistem, a neispravnost tipa „pogrešno očitavanje senzora temperature u usisnom kolektoru“ je izostavljena iz laboratorijskog ispitivanja, da bi se umanjio rizik od krupnijih štetnih

posledica za ispitivani motor usled rada na velikom broju neregularnih režima. Svakako da bi uzimanje u obzir i ovih neispravnosti na stvarnom motoru doprinelo kompletnijoj eksperimentalnoj verifikaciji simulatora neispravnosti, pa bi to mogao da bude zadatak u nekim budućim istraživanjima. Osim toga, bilo bi poželjno da se u nekom narednom istraživanju ispita i kako se ponaša stvarni motor sa masenim protokomerom vazduha kada u njegovom usisnom sistemu postoji neka od ovde razmatranih neispravnosti.

Drugi važan cilj ovog istraživanja bio je da se razvije dijagnostički model koji bi mogao uspešno da dijagnostikuje gore pomenute tipove neispravnosti, pri radu motora na ustaljenom režimu. Tokom izrade ove disertacije ovaj cilj je u potpunosti realizovan, čime je potvrđena naučna hipoteza postavljena na početku istraživanja.

U okviru ove disertacije predložen je novi dijagnostički koncept, u kojem je dijagnostički model razvijan na velikom broju podataka koji se dobijaju od različitih senzora tokom funkcionisanja motora. Dijagnostički model sačinjava osam nezavisnih neuronskih mreža (četiri za rad motora u zatvorenoj i četiri za rad motora u otvorenoj sprezi sa lambda-sondom). Pri tome, svaka od četiri neuronske mreže za rad u odgovarajućem lambda-režimu, aktivna je u jednom od četiri razmatrana stepena prenosa prilikom kretanja vozila. Ovaj dijagnostički koncept je primenjen na podatke dobijene pomoću simulatora neispravnosti, pri čemu je realizovan tzv. simulacioni dijagnostički model, a onda i na podatke dobijene laboratorijskim ispitivanjem stvarnog motora, čime je razvijen i laboratorijski dijagnostički model. U oba slučaja dobijeni su veoma dobri rezultati, čak i pri rešavanju najtežeg dijagnostičkog zadatka: prepoznavanje nepoznatih neispravnosti na nepoznatim režimima rada motora. Pored toga, s obzirom da se razvijeni dijagnostički model zasniva na neuronskim mrežama, treba pomenuti da je njegova prednost u odnosu na analitičke modele u tome, što se veoma brzo izvršavaju, a to ih čini pogodnim za eventualnu primenu na stvarnim vozilima. Još jedna dobra strana razvijenog dijagnostičkog modela je što se njegovi parametri (težinski koeficijenti neuronske mreže) određuju automatski tokom obuke, tako da nije potreban poseban postupak parametrizacije modela.

Naravno, postoje i određeni nedostaci predloženog dijagnostičkog koncepta. Neki od tih nedostataka su, u suštini, nedostaci neuronske mreže, kao osnovne strukturne jedinice razvijenog dijagnostičkog modela. Jedan od osnovnih nedostataka je odsustvo eksplicitne veze između parametara modela i fizičkih parametara motora, što u značajnoj meri oduzima fleksibilnost dijagnostičkom modelu. To znači da je neophodno izvršiti ponovnu obuku modela, u slučaju izmene nekih fizičkih parametara sistema (npr. ako se posmatrani motor želi ugraditi u neki drugi automobil, čiji parametri nisu korišćeni pri prvobitnoj obuci modela). Pored toga, kao nedostatak predloženog dijagnostičkog koncepta može se smatrati i potreba za obimnim i intenzivnim radom na prikupljanju podataka za obuku i testiranje modela.

Pored gore navedenih nedostataka koji su vezani direktno za neuronske mreže, postoje i neke slabosti i ograničenja vezana za konkretne dijagnostičke modele, razvijene u ovom radu. To se pre svega može reći za laboratorijski dijagnostički model, jer njegovo testiranje nije izvedeno u tako velikom broju radnih tačaka motora, kao što je to slučaj sa simulacionim dijagnostičkim modelom. Ovo se, pre svega, odnosi na činjenicu da nije testiran uticaj promene pojedinih parametara vozila na dijagnostičke performanse laboratorijskog modela, pa je ostalo nepoznato kako bi se taj model ponašao u pomenutim novonastalim uslovima. Zbog toga bi trebalo da ova nepoznanica dobije odgovor u nekim budućim istraživanjima.

Zajednički nedostatak oba razvijena dijagnostička modela odnosno nedostatak celog koncepta je što su ograničeni na ustaljeni rad motora. S obzirom da rad automobilskog motora karakteriše relativno česta promena režima rada, postavlja se pitanje dijagnostikovanja neispravnosti na prelaznim režimima. Predloženi koncept ne nudi odgovor na to pitanje, ali bi rešavanje ovog problema moglo da bude u fokusu nekih



obimnijih istraživanja u vremenu koje dolazi. Pored ograničenih mogućnosti modela po pitanju ustaljenog režima rada, sličnim nedostatkom bi se moglo smatrati i ograničenje na relativno mali deo ukupnog radnog dijapazona motora. Pomenuti nedostatak bi mogao biti otklonjen u nekom narednom istraživanju, koje bi bilo izvodljivo kada se značajnije unaprede performanse softvera za razvoj neuronskih mreža.

Dalje, sledeće ograničenje koje je prisutno u predloženom dijagnostičkom konceptu jeste pretpostavka, da u usisnom sistemu motora ne mogu postojati dve ili više neispravnosti istovremeno. Ovo jeste prilično realna pretpostavka, ali se ipak može desiti da u usisnom sistemu postoji više neispravnosti. Ako do toga dođe, to znači da nešto nije u redu sa dijagnostičkim sistemom, koji je trebalo da reaguje već na pojavu prve neispravnosti. Bilo kako bilo, i ovo bi mogla da bude prilično inspirativna tema za neko sledeće istraživanje.

Veoma zanimljivo bi moglo da bude i istraživanje mogućnosti konkretne realizacije predloženog dijagnostičkog koncepta u nekom realnom OBD sistemu. Time bi se dobila jasnija slika o praktičnoj upotrebljivosti ovog koncepta, a u slučaju nedostataka, koji bi se tom prilikom zapazili, otvorile bi se mogućnosti za nova istraživanja.

Bilo bi interesantno primeniti ovde opisani dijagnostički koncept i na prehranjivane motore, za koje je karakteristično da, u slučaju postojanja nepredviđenog otvora u usisnom sistemu, vazduh pri nekim režimima može ulaziti u usisni sistem, a pri nekim drugim režimima može iz njega izlaziti. Neko naredno istraživanje bi moglo da utvrdi kako bi se predloženi dijagnostički koncept pokazao kod takvih motora.

Jedan od ozbiljnih zadataka u narednim istraživanjima moglo bi da bude proučavanje ponašanja dizel motora pri pojavi nekih neispravnosti u usisnom sistemu, kao i razvoj odgovarajućeg dijagnostičkog modela koji bi te neispravnosti uspešno prepoznao.

## 9 LITERATURA

- [1] Nyberg, M.: Model based diagnosis of both sensor faults and leakage in the air-intake system of an SI engine. SAE Paper 1999-01-0860 (1999)
- [2] Isermann, R.: Fault Diagnosis Applications. Springer-Verlag, Berlin, Heidelberg (2011)
- [3] Sangha, M.: Intelligent fault diagnosis for automotive engines and real data evaluation, PhD Thesis., Liverpool (2008)
- [4] Gertler, J.: Survey of Model-Based Failure Detection and Isolation in Complex Plants. IEEE Control System Magazine, Vol 8(6), 3-11 (1998)
- [5] Venkatasubramanian, V., Rengaswamy, R., Yin, K., Kavuri, S.: A review of process fault detection and diagnosis (Part I: Quantitative model-based methods). Computers and Chemical Engineering, 293-311 (2003)
- [6] Venkatasubramanian, V., Rengaswamy, R., Kavuri, S.: A review of process fault detection and diagnosis (Part II: Qualitative models and search strategies). Computers and Chemical Engineering, 313-326 (2003)
- [7] Venkatasubramanian, V., Rengaswamy, R., Kavuri, S., Yin, K.: A review of process fault detection and diagnosis (Part III: Process history based methods). Computers and Chemical Engineering, 327-346 (2003)
- [8] Patton, R., Uppal, F., Lopez-Toribio, C.: Soft Computing Approaches to Fault Diagnosis for Dynamic Systems: a Survey. In : IFAC symposium on fault detection, supervision and safety for technical processes, Budapest, Hungary, pp.298-311 (2000)
- [9] Klinar, I., Torović, T., Nikolić, N., Antonić, Ž.: Neki aspekti dijagnostike motora SUS. Traktori i pogonske mašine VIII(3), 44-50 (2003)
- [10] Klinar, I., Furman, T.: Savremeni dijagnostički sistemi i pogodnost za dijagnostiku motora SUS. Traktori i pogonske mašine VIII(3), 51-57 (2003)
- [11] Jackson, P.: Introduction to Expert Systems. Addison-Wesley, Boston (1990)
- [12] Durkin, J., Durkin, J.: Experts Systems: Design and Development. Prentice Hall, New Jersey (1998)
- [13] Angeli, C.: Diagnostic Expert Systems: From Expert's Knowledge to Real-Time Systems. In : Advanced Knowledge Based Systems: Models, Applications & Research. Technomathematics Research Foundation, Kolhapur, India (2010) 50-73
- [14] Chrissanthi, A.: Online expert systems for fault diagnosis in technical processes. Expert Systems, 115-132 (2008)
- [15] Celik, M., Bayir, R.: Fault Detection in internal combustion engines using fuzzy logic. Proceedings of the Institution of Mechanical Engineers, Part D: Journal of Automobile Engineering, 579-587 (2007)
- [16] Chow, E., Wilsky, A.: Analytical Redundancy and the Design of Robust Failure Detection Systems. IEEE Transactions on Automatic Control, 603-614 (1984)
- [17] Patan, K.: Artificial Neural Networks for the Modelling and Fault Diagnosis of Technical processes. Springer Verlag, Berlin, Heidelberg (2008)
- [18] Isermann, R.: Fault-Diagnosis Systems - An Introduction from Fault Detection to

- Fault Tolerance. Springer-Verlag, Berlin, Heidelberg, New York (2006)
- [19] Korbicz, J., Koscielny, J., Kowalczyk, Z., Cholewa, W.: Fault Diagnosis - Models, Artificial Intelligence, Applications. Springer-Verlag, Berlin, Heidelberg (2004)
- [20] Rizzoni, G., Hampo, R.: Real time detection filters for on-board diagnosis of incipient failures. SAE Paper No. 890763 (1989)
- [21] Rizzoni, G., Min, P.: Detection of sensors failures in automotive engines. IEEE Transactions on Vehicular Technology, 487-500 (1991)
- [22] Rizzoni, G., Azzoni, P., Minelli, G.: On-board diagnosis of emission control system malfunctions in electronically controlled spark-ignition engines. In : Proceedings of the American Control Conference, pp.1790-1795 (1993)
- [23] Gertler, J., Costin, M., Fang, X., Hira, R., Kowalczyk, Z., Luo, Q.: Model-based on-board fault detection and diagnosis for automotive engines. In : IFAC Fault Detection, Supervision and Safety for Technical Processes, Baden-Baden, Germany, pp.503-508 (1991)
- [24] Gertler, J., Costin, M., Fang, X., Hira, R., Kowalczyk, Z., Luo, Q.: Model-based on-board fault detection and diagnosis for automotive engines. Control Engineering Practice, 3-17 (1993)
- [25] Gertler, J., Costin, M., Fang, X., Kowalczyk, Z., Kunwer, M., Monajemy, R.: Model-based diagnosis of automotive engines - algorithm development and testing on a production vehicle. IEEE Transactions on Control Systems Technology, 61-69 (1995)
- [26] Nyberg, M., Nielsen, L.: Model based diagnosis for the air intake system of SI engine. Journal of commercial vehicles, SAE Transaction, Vol. 106, 9-20 (1997)
- [27] Nyberg, M., Perkovic, A.: Model based diagnosis of leaks in the air intake system of an SI engine. SAE paper No. 980514 (1998)
- [28] Conatser, R., Wagner, J., Ganta, S., Walker, I.: Diagnosis of automotive electronic throttle control systems. Control Engineering Practice, 23-30 (2004)
- [29] Franchek, M., Buehler, P., Makki, I.: Intake air path diagnostics for internal combustion engines. Journal of Dynamic Systems Measurement and Control, 32-40 (2007)
- [30] Sengupta, S., De, S., Bhattacharya, A., Mukhopadhyay, S., Deb, A.: Fault Detection of Air Intake Systems of SI Gasoline Engines using Mean Value and Within Cycle Models. In : 5th Annual Conference on Automation Science and Engineering, Bangalore, pp.361-366 (2009)
- [31] Ceccarelli, R., Canudas de Wit, C., Moulin, P., Sciarretta, A.: Model-based Adaptive Observers for Intake Leakage Detection in Diesel Engine. In : American Control Conference, Sent Louis, USA (2009)
- [32] Salehi, R., Aria, A., Gholamreza, V.: Model-based air leak detection for turbocharged gasoline engines without a hot-film air mass flow meter sensor. Proceedings of the Institution of Mechanical Engineers, Part D: Journal of Automobile Engineering, 1-18 (2014)
- [33] Sangha, M., Yu, D., Gomm, B.: On-board Monitoring and Diagnosis for Spark Ignition Engine Air Path via Adaptive Neural Networks. Proceedings of the Institution of Mechanical Engineers, Part D: Journal of Automobile Engineering, 1641-1655 (2006)
- [34] Sangha, M., Yu, D., Gomm, B.: Robustness Assessment and Adaptive FDI for Car Engine. International Journal of Automation and Computing, 109-118 (2008)

- [35] Hamad, A., Yu, D., Gomm, B., Sangha, M.: Radial basis function neural network in fault detection of automotive engines. *International Journal of Engineering, Science and Technology*, 1-8 (2010)
- [36] Yu, D., Hamad, A., Gomm, B., Sangha, M.: Dynamic fault detection and isolation for automotive engine air path by independent neural network model. *International Journal of Engine research*, 87-100 (2014)
- [37] Chen, P.-C.: A novel diagnostic system for gasoline-engine leakage detection. *Proceedings of the Institution of Mechanical Engineers, Part D: Journal of Automobile Engineering*, 673-685 (2011)
- [38] Sangha, M., Yu, D., Gomm, B.: Sensor fault detection, isolation, accommodation and unknown fault detection in automotive engine using AI. *International Journal of Engineering, Science and Technology*, 53-65 (2012)
- [39] Nyberg, M.: Automatic design of diagnosis systems with application to an automotive engine. *Control Engineering Practice*, 993-1005 (1999)
- [40] Nyberg, M.: Evaluation of test quantities for leakage diagnosis in the air path of an automotive engine. In : *IFAC fault detection, supervision and safety for technical processes*, Budapest, Hungary, pp.143-148 (2000)
- [41] Nyberg, M.: A general framework for fault diagnosis based on statistical hypothesis testing. In : *American association for artificial intelligence*, pp.1-8 (2001)
- [42] Nyberg, M., Stutte, T.: Model based diagnosis of the air path of an automotive diesel engine. *Control Engineering Practice*, 513-525 (2004)
- [43] Capriglione, D., Ligouri, C., Pianese, C., Pietrosanto, A.: Analytical redundancy for sensor fault isolation and accommodation in public transportation vehicles. *IEEE Transactions on Instrumentation and Measurement*, 993-999 (2004)
- [44] Isermann, R., Schwarte, A.: Fault diagnosis of a turbocharged diesel engine with dynamic neural networks and parity methods. In : *Proceedings of the 6th IASTED International Conference on Intelligent Systems and Control*, Honolulu, Hawaii, USA, pp.283-288 (2004)
- [45] Kimmich, F., Schwarte, A., Isermann, R.: Fault detection for modern diesel engines using signal and proces model based methods. *Control Engineering Practice*, 189-203 (2005)
- [46] Chen, P.: Neuro-fuzzy-based fault detection of the air flow sensor of an idling gasoline engine. *Proceedings of the Institution of Mechanical Engineers, Part D: Journal of Automobile Engineering*, 511-524 (2005)
- [47] Antory, D.: Application of a data-driven monitoring technique to diagnose air leaks in an automotive diesel engine: A case study. *Mechanical Systems and Signal Processing*, 795-808 (2007)
- [48] Wang, X., McDowell, N., Kruger, U., McCullough, G., Irwin, G.: Semi-physical Neural Network Model in Detecting Engine Transient Faults using the Local Approach. In : *Proceedings of the 17th IFAC World Congress*, Seoul, South Korea, pp.7086-7090 (2008)
- [49] Qadeer, A., Aamer, I., Imtiaz, T., Khubaib, A.: Gasoline Engine Intake Manifold Leakage Diagnosis/Prognosis Using Hidden Markov Model. *International Journal of Innovative Computing, Information and Control*, 4661-4674 (2012)
- [50] Glesner, M., Pöschmüller, W.: *Neurocomputers: An Overview of Neural Networks in VLSI*. Chapman & Hall, London (1994)
- [51] Petriu, E., Watanabe, K., Yeap, T.: *Applications of Random-Pulse Machine Concept to*

- Neural Network Design. Instrumentation and Measurement, IEEE Transactions, 665-669 (1996)
- [52] Haykin, S.: Neural Networks: A Comprehensive Foundation. MacMillan College Publishing, New York (1994)
- [53] Samarasinghe, S.: Neural Networks for Applied Sciences and Engineering: From Fundamentals to Complex Pattern recognition. Auerbach Publications, Boca Raton (2007)
- [54] Fausett, L.: Fundamentals of Neural Networks—Architectures, Algorithms and Applications. Prentice Hall, New Jersey (1994)
- [55] Gurney, K.: An introduction to neural networks. UCL Press Limited, London (2005)
- [56] Hagiwara, M.: Removal of hidden units and weights for backpropagation. In : International Joint Conference on Neural Networks, Nagoya, pp.351-354 (1993)
- [57] Castellano, G., Fanelli, A., Pelillo, M.: An Iterative Pruning Algorithm for Feedforward Neural Networks. IEEE Transactions on Neural Networks, 519-531 (1997)
- [58] Tamura, S., Tateishi, M.: Capabilities of a four-layered feedforward neural network: four layers versus three. IEEE Transactions on Neural Networks, 251-255 (1997)
- [59] Reed, R., Marks II, J.: Neural Smithing: Supervised Learning in Feedforward Artificial Neural Networks. The MIT Press, Cambridge (1999)
- [60] Hornik, K., Stinchcombe, M., White, H.: Multilayer feedforward networks are universal approximators. Neural Networks, 359-366 (1989)
- [61] Stinchcombe, M., White, H.: Universal Approximation Using Feedforward Networks with Non-sigmoid Hidden Layer Activation Function. In : International Joint Conference on Neural Networks, New York, pp.613-617 (1989)
- [62] Bishop, C.: Neural Networks for Pattern Recognition. Oxford University Press Inc., New York (2005)
- [63] Chester, D.: Why two hidden layers are better than one. In : International Joint Conference on Neural Networks, Washington DC, pp.265-268 (1990)
- [64] Sontag, E.: Feedback stabilization using two-hidden-layer nets. IEEE Transactions on Neural Networks, 981-990 (1992)
- [65] LeCun, Y., Boser, B., Denker, J., Hederson, D., Howard, R., Hubbard, W., Jackel, L.: Backpropagation Applied to Handwritten Zip Code Recognition. Neural Computation, 541-551 (1989)
- [66] Nascimento, C.: Artificial Neural Networks in Control and Optimization, PhD dissertation., Manchester (1994)
- [67] Rosenblatt, F.: The perceptron: A probabilistic model for information storage and organization in the brain. Psychological Review, 386-408 (1958)
- [68] Hagan, M., Demuth, H., Beale, M.: Neural Network Design. Thomson Asia Pre Ltd (2002)
- [69] Rumelhart, D., Hinton, G., Williams, R.: Learning Internal Representations by Error Propagation. In : Parallel Distributed Processing: Explorations in the Microstructure of Cognition, Volume I: Foundations. MIT Press, Cambridge, USA (1986) 318-362
- [70] Kröse, B., van der Smagt, P.: An Introduction to Neural Networks. University of Amsterdam, Amsterdam (1996)
- [71] Beale, M., Hagan, M., Demuth, H.: Neural Network Toolbox 7, User's Guide. The MathWorks, Inc., Natick, USA (April 2011)
- [72] Guzzella, L., Onder, C.: Introduction to Modeling and Control of Internal Combustion

- Engine Systems. Springer-Verlag, Heidelberg (2010)
- [73] Isermann, R., Schaffnit, J., Sinsel, S.: Hardware-in-the-loop simulation for the design and testing of engine-control systems. *Control Engineering Practice*, 643-653 (1999)
- [74] "dSpace": Automotive Solutions, Systems and Applications. In: <http://files.hanser-tagungen.de>. (Accessed 2009) Available at: [http://files.hanser-tagungen.de/docs/20111025131724\\_dSPACE Automotive Solutions E.pdf](http://files.hanser-tagungen.de/docs/20111025131724_dSPACE%20Automotive%20Solutions%20E.pdf)
- [75] Fathy, H., Filipi, Z., Hagen, J., Stein, J.: Review of Hardware-in-the-Loop Simulation and Its Prospects in the Automotive Area. In : *Proceedings of International Conference of Modeling and Simulation for Military Applications*, p.62280E (2006)
- [76] Yang, X., Zhu, G.: A Mixed Mean-Value and Crank-Based Model of a Dual-Stage Turbocharged SI Engine for Hardware-In-the-Loop Simulation. In : *American Control Conference*, Baltimore, pp.3791-3796 (2010)
- [77] Benz, M., Onder, C., Guzzella, L.: Engine Emission Modeling Using a Mixed Physics and Regression Approach. *Journal of Engineering for Gas Turbines and Power*, doi:10.1115/1.3204510 (2010)
- [78] Qu, Z., Ma, M., Zhao, F.: An Online Crank-Angle-Resolved Mean-Value Combustion Model of Gasoline Engines Including Effects of Cycle Initial States. *SAE Technical Paper*, doi:10.4271/2012-01-0129 (2012)
- [79] Casoli, P., Gambarotta, A., Pompini, N., Caiazzo, U., Lanfranco, E., Palmisano, A.: Development and validation of a "crank-angle" model of an automotive turbocharged Engine for HiL Applications. *Energy Procedia*, 839-848 (2014)
- [80] Hendricks, E., Sorenson, S.: Mean Value Modelling of Spark Ignition Engines. *SAE Technical Paper* 900616 (1990)
- [81] Müller, M., Hendricks, E., Sorenson, S.: Mean Value Modelling of Turbocharged Spark Ignition Engines. *SAE Technical Papers*, doi:10.4271/980784. (1998)
- [82] Jung, M.: Mean-Value Modelling and Robust Control of the Airpath of a Turbocharged Diesel Engine, PhD thesis., University of Cambridge, Cambridge (2003)
- [83] Eriksson, L., Nielsen, L., Brugard, J., Bergström, J., Pettersson, F., Andersson, P.: Modeling of a Turbocharged SI Engine. *Annual Reviews in Control*, 129-137 (2002)
- [84] Jazayeri, S., Rad, M., Azadi, S.: Development and Validation for Mean Value Engine Models. In : *ASME 2005 Internal Combustion Engine Division Fall Technical Conference*, Ottawa, pp.19-28 (2005)
- [85] Schulze, T., Wiedemeier, M., Schuette, H.: Crank Angle - Based Diesel Engine Modeling for Hardware-in-the-Loop Applications with In-Cylinder Pressure Sensors. *SAE Technical Paper*, doi:10.4271/2007-01-1303 (2007)
- [86] Shaver, G., Roelle, M., Gerdes, C.: Modeling cycle-to-cycle dynamics and mode transition in HCCI engines with variable valve actuation. *Control Engineering Practice*, 213-222 (2006)
- [87] Vasu, J., Deb, A., Mukhopadhyay, S., Pattada, K.: Development and Validation of an MVEM from an SI-Engine based WCCM. In : *International Conference on Modelling, Identification and Control*, Shanghai, pp.52-57 (2011)
- [88] Hendricks, E.: Mean Value Modelling of Large Turbocharged Two-Stroke Diesel Engines. *SAE Technical Paper*, doi:10.4271/890564 (1989)
- [89] Carlsson, J., Fredriksson, J.: Cylinder-by-cylinder engine models vs mean value engine models for use in powertrain control systems. *SAE Technical Paper* 1999-01-

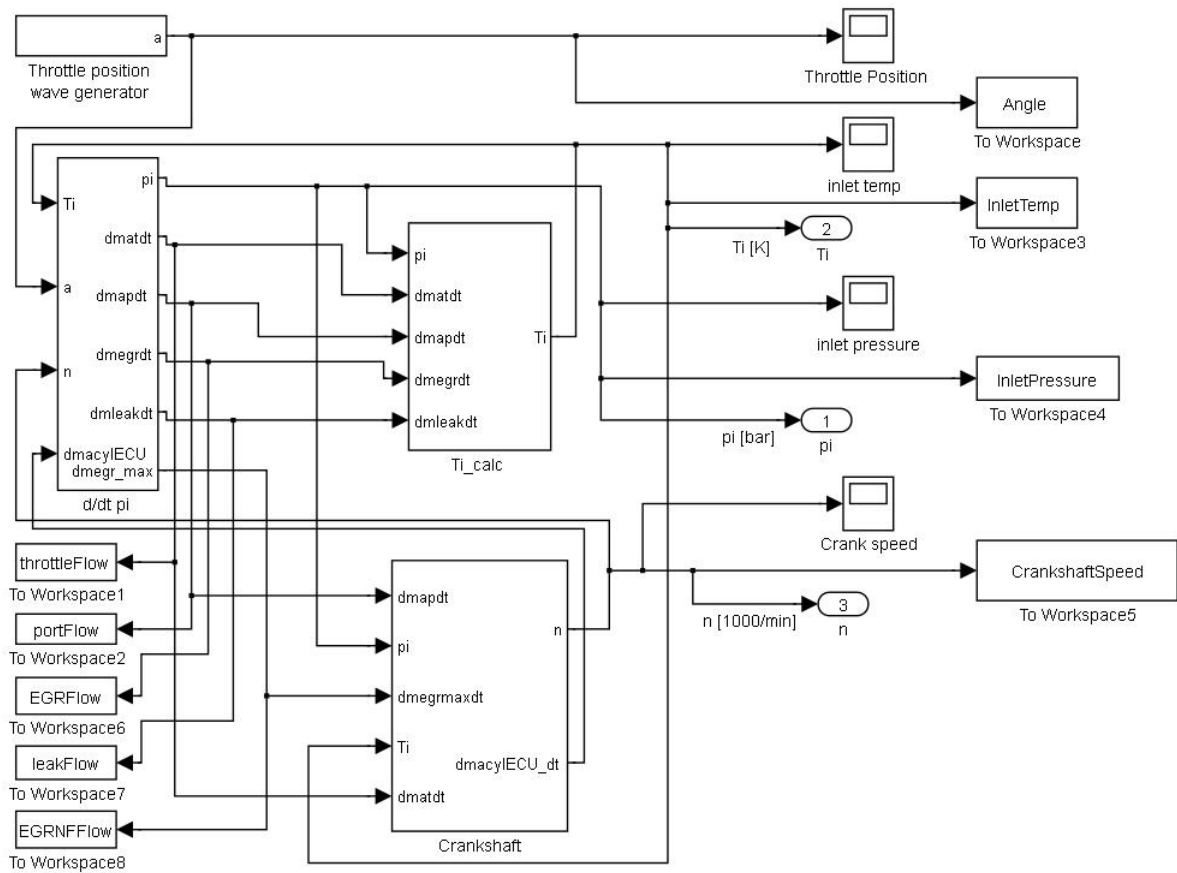
- 0906, doi:10.4271/1999-01-0906 (1999)
- [90] He, Y., Lin, C.: Development and Validation of a Mean Value Engine Model for Integrated Engine and Control System Simulation. SAE Technical Paper 2007-01-1304, doi. 10.4271/2007-01-1304 (2007)
- [91] Hendricks, E., Vesterholm, T., Kaidantzis, P., Rasmussen, P., Jensen, M.: Nonlinear Transient Fuel Film Compensation (NTFC). SAE Technical Paper, doi:10.4271/930767 (1993)
- [92] Hendricks, E., Chevalier, A., Jensen, M., Sorenson, S., Trumpy, D., Asik, J.: Modelling of the Intake Manifold Filling Dynamics. SAE Technical Paper, doi. 10.4271/960037 (1996)
- [93] Fons, M., Müller, M., Chevalier, A., Vigild, C., Hendricks, E., Sorenson, S.: Mean Value Engine Modelling of an SI Engine with EGR. SAE Technical Paper, doi. 10.4271/1999-01-0909 (1999)
- [94] Chevalier, A., Müller, M., Hendricks, E.: On the Validity of Mean Value Engine Models During Transient Operation. SAE Technical Paper, doi:10.4271/2000-01-1261 (2000)
- [95] Heywood, J.: Internal Combustion Engine Fundamentals. McGraw-Hill, New York (1988)
- [96] Hendricks, E., Engler, D., Fam, M.: A Generic Mean Value Engine Model for Spark Ignition Engines. In : 41st Simulation Conference, SIMS 2000, Lyngby (2000)
- [97] White, F.: Fluid Mechanics, 4th Edition. McGraw-Hill, New York (1999)
- [98] Aquino, C.: Transient A/F Control Characteristics of the 5 Liter Central Fuel Injection Engine. SAE Technical Paper, doi. 10.4271/810494 (1981)
- [99] Dobner, D., Fruechte, R.: An Engine Model for Dynamic Engine Control Development. In : American Control Conference, San Francisco, USA, pp.73-78 (1983)
- [100] Moskwa, J., Hedrick, J.: Automotive Engine Modeling for Real Time Control Applications. In : American Control Conference, Minneapolis, USA, pp.341-346 (1987)
- [101] Abida, J., Claude, D.: Spark Ignition Engines and Pollution Emission: New Approaches in Modelling and Control. International Journal of Vehicle Design, 494-508 (1994)
- [102] Taylor, C., Taylor, E.: The Internal Combustion Engine. International Textbook, Scranton (1970)
- [103] Sangha, M., Gomm, B., Yu, D.: Neural network fault classification of transient data in an automotive engine air path. International Journal of Modelling, Identification and Control, 148-155 (2008)
- [104] Basshuysen, R., Schäfer, F.: Handbuch Verbrennungsmotor: Grundlagen, Komponenten, Systeme, Perspektiven. Wieteg, Wiesbaden (2007)
- [105] Stojić, B.: Teorija kretanja drumskih vozila - nastavni materijali. In: Teorija kretanja drumskih vozila. (Accessed Februar 2012) Available at: <http://tkdv.files.wordpress.com/2012/02/teorija-kretanja-drumskih-vozila-skripta.pdf>
- [106] Simić, D.: Motorna vozila. Naučna knjiga, Beograd (1988)
- [107] Rajamani, R.: Vehicle Dynamics and Control. Springer, New York (2006)
- [108] Mitschke, M., Wallentowitz, H.: Dynamik der Kraftfahrzeuge. Springer-Verlag, Berlin

- Heidelberg New York (2004)
- [109] Cengel, Y., Boles, M.: Thermodynamics - An Engineering Approach. McGraw-Hill, New York (2002)
- [110] Nikolić, N., Torović, T., AntoniĆ, Ź., Dorić, J.: Investigation of Usage Possibilities of a Mean Value Engine Model in Simulation of IC Engine Intake System Faults. In : International Conference on Innovative Technologies, INTECH 2012, Rijeka, Croatia, pp.125-129 (2012)
- [111] Halderman, J.: Automotive Technology: Principles, Diagnosis and Service. Pearson Prentice Hall, New Jersey (2009)
- [112] Nikolić, N., Torović, T., AntoniĆ, Ź., Dorić, J.: Simulacija neispravnosti u usisnom sistemu pomoću osrednjenog modela motora. Traktori i pogonske mašine, 105-111 (2012)
- [113] Ribbens, W.: Understanding Automotive Electronics. Newnes, Boston London New York (2003)
- [114] Warren, M.: Motor Magazine. In: Motor Magazine. (Accessed Decembar 2001) Available at: [http://www.motor.com/magazine/pdfs/122001\\_06.pdf](http://www.motor.com/magazine/pdfs/122001_06.pdf)
- [115] MAF Test part 1. In: YouTube. (Accessed Januar 9, 2012) Available at: <http://www.youtube.com/watch?v=D9-Ug9c9Xl4>
- [116] Dorić, J., Klinar, I., Raspopović, N., Dakić, B., Nikolić, N., AntoniĆ, Ź.: Novo eksperimentalno postrojenje za ispitivanje motora SUS, tehničko rešenje., Novi Sad (2012)
- [117] Dorić, J.: Unapređenje efektivnosti motora SUS primenom nekonvencionalnog klipnog mehanizma, doktorska disertacija., Novi Sad (2012)
- [118] Russell, J.: Versatile Engine Management Systems. In: VEMS user guide. (Accessed December 16, 2012) Available at: <http://www.vems.us/downloads/V3installandsetupmanual.pdf>
- [119] CARL SCHENCK AG: Wierbelstrombremse W230, Bedienungseinleitung., Darmstadt (1977)
- [120] Norton Electronic: HEDSS encoders - shaft encoders. In: Norton Electronic. Available at: <http://www.nortonelectronic.com/en/0/hedss-encoders/shaft-encoders/isc3806.html>
- [121] VEMS group: VEMS shop catalog - MAP sensor. In: VEMS shop catalog. Available at: <https://shop.vems.hu/catalog/sensor-p-89.html?osCsid=a4ddad26c820164c37621a535ff459f7>
- [122] Lambda sensor LSU 4.2. In: Bosch motorsport. Available at: [http://www.bosch-motorsport.de/media/catalog\\_resources/Lambda\\_Sensor\\_LSU\\_42\\_Datasheet\\_51\\_en\\_2779111435pdf.pdf](http://www.bosch-motorsport.de/media/catalog_resources/Lambda_Sensor_LSU_42_Datasheet_51_en_2779111435pdf.pdf)

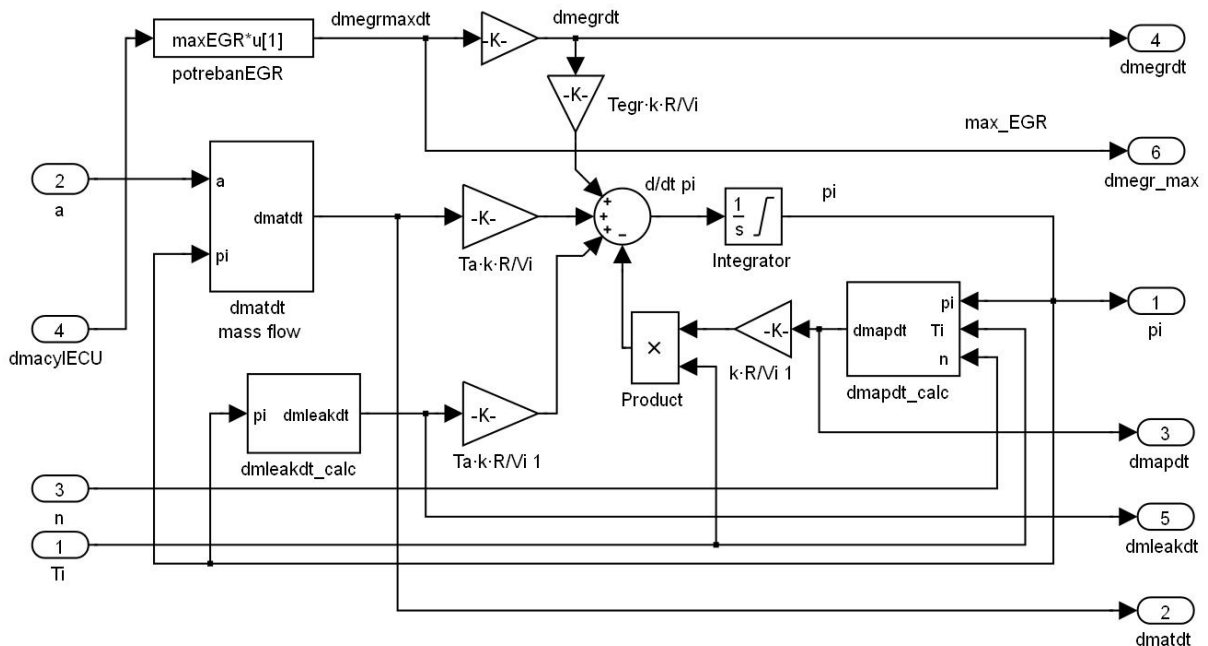


## **10 PRILOZI**

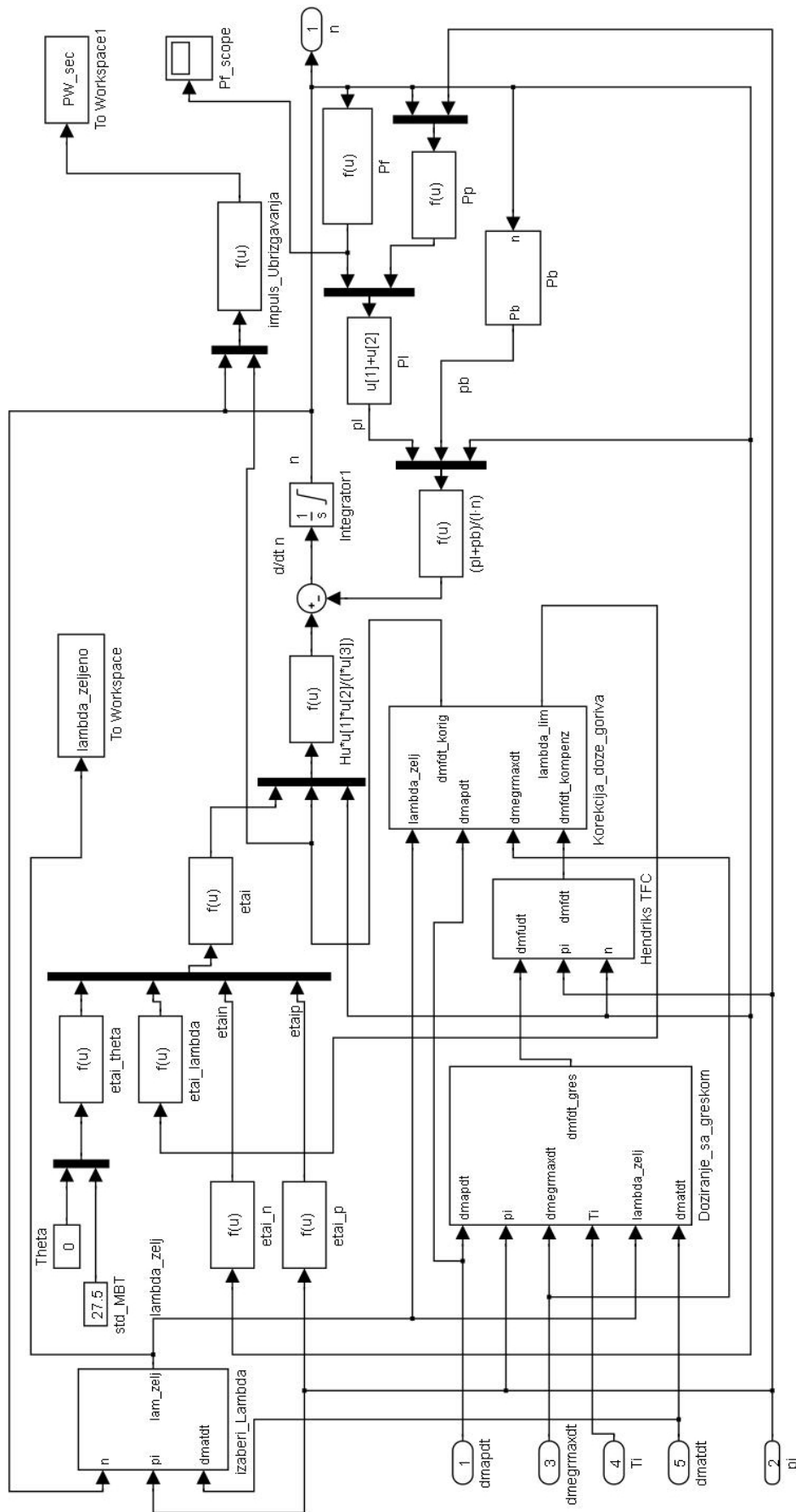
# P1 Najvažniji moduli Simulink modela NSimulator



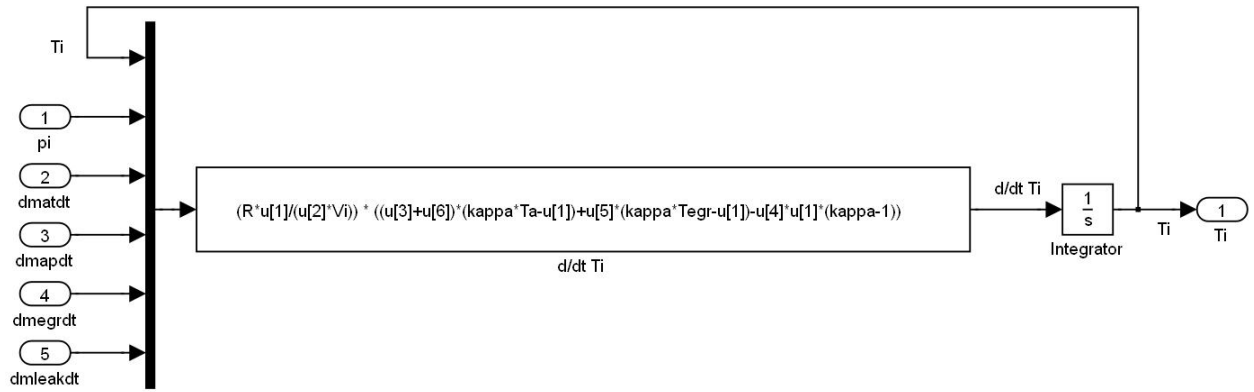
Sl. 10.1 Šema prvog nivoa kompletnog modela NSimulator



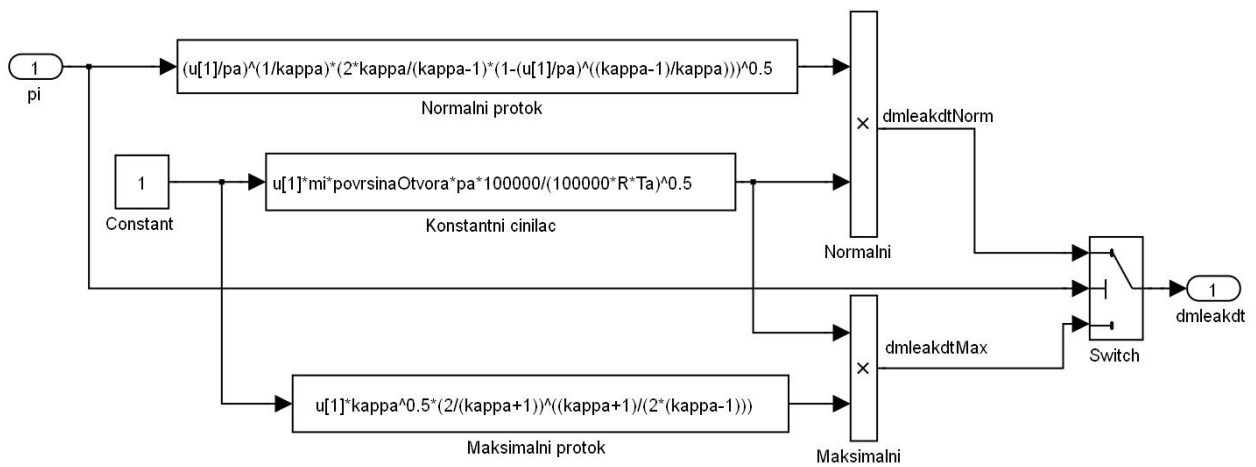
Sl. 10.2 Izračunavanje pritiska u usisnom kolektoru (modul „d/dt pi“ sa Sl. 10.1)



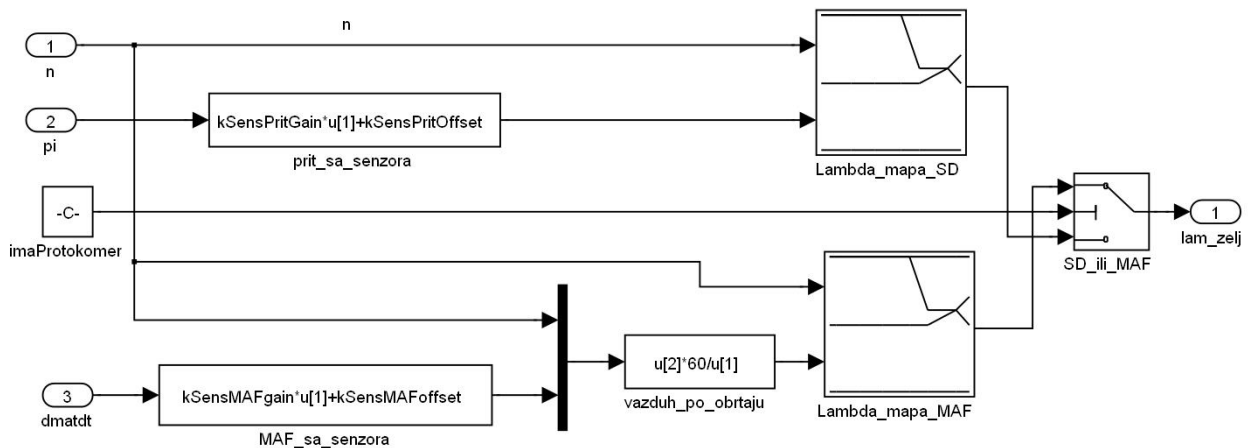
Sl. 10.3 Određivanje broja obrtaja kolenastog vratila (modul „Crankshaft“ sa Sl. 10.1)



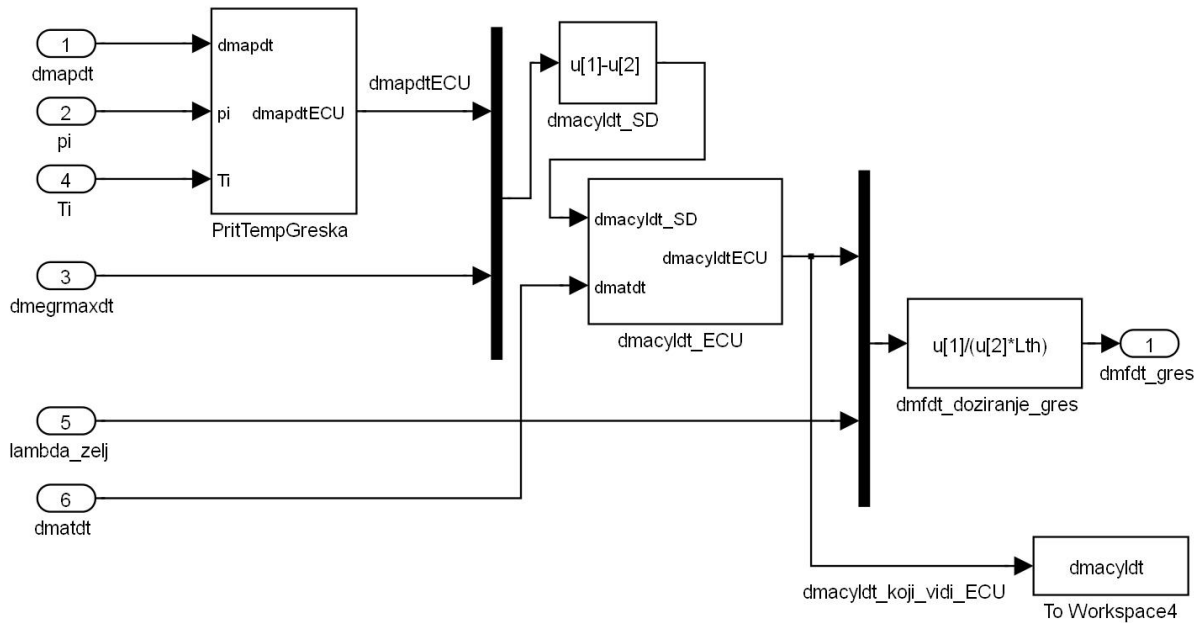
Sl. 10.4 Izračunavanje temperature u usisnom kolektoru (modul „Ti\_calc“ sa Sl. 10.1)



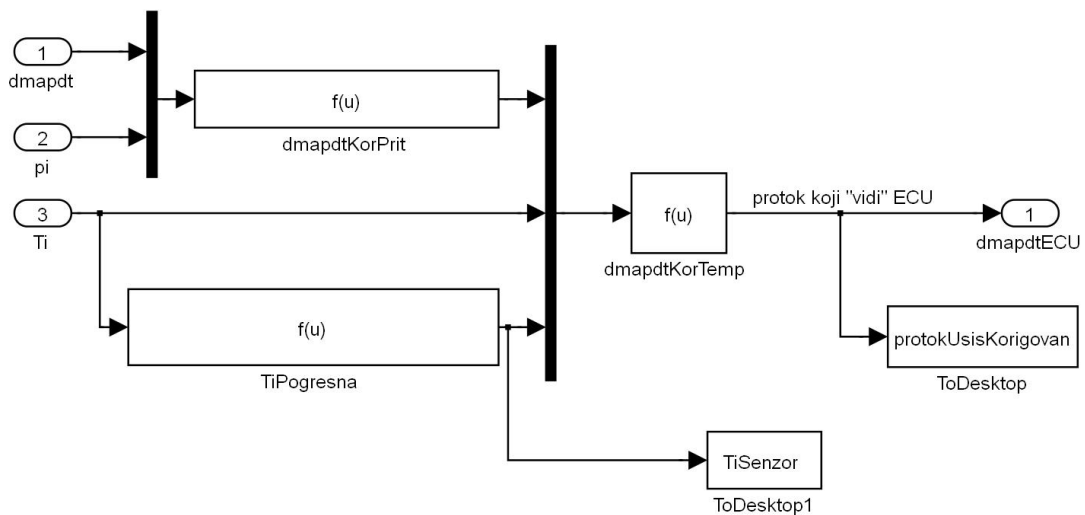
Sl. 10.5 Određivanje masenog protoka „falš“ vazduha (modul „dmleakdt\_calc“ sa Sl. 10.2)



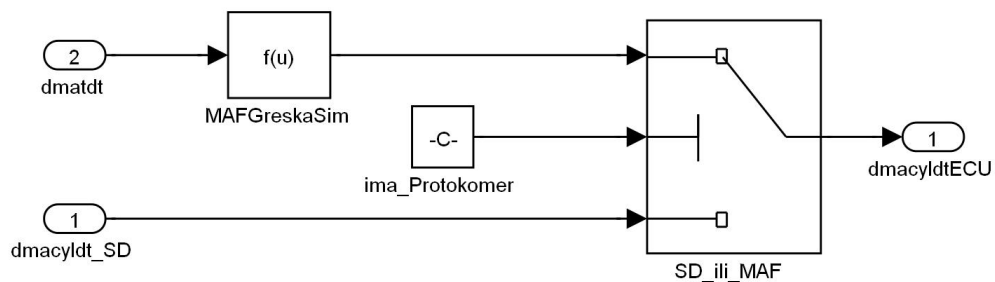
Sl. 10.6 Izbor željene vrednosti koeficijenta viška vazduha u zavisnosti od režima rada motora (modul „Izaberi\_lambda“ sa Sl. 10.3)



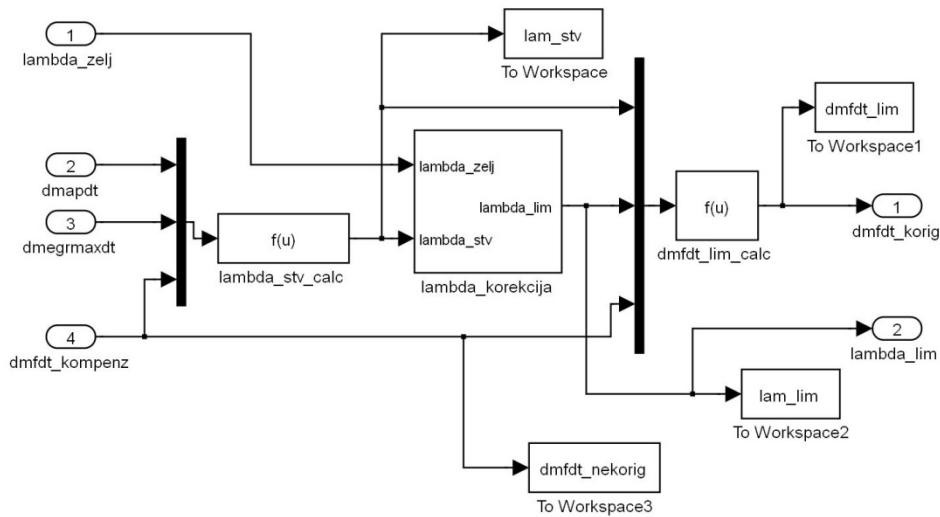
Sl. 10.7 Određivanje potrebne količine goriva pri postojanju neispravnosti u usisnom sistemu (modul „Doziranje\_sa\_greskom“ sa Sl. 10.3)



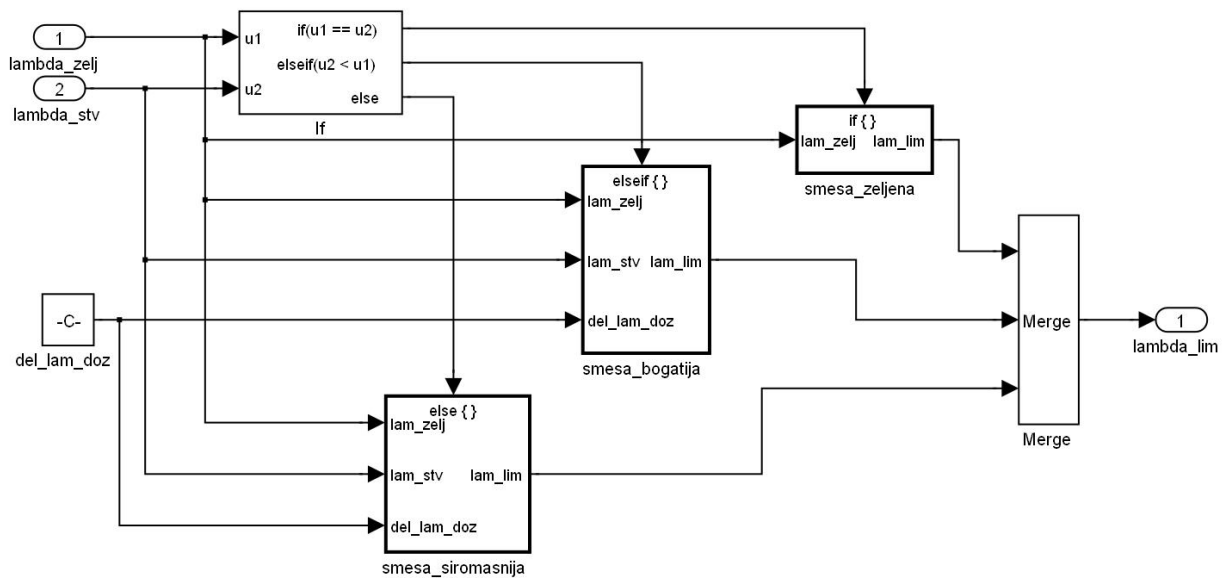
Sl. 10.8 Simuliranje grešaka senzora pritiska i temperature u usisnom kolektoru (modul „PritTempGreska“ sa Sl. 10.7)



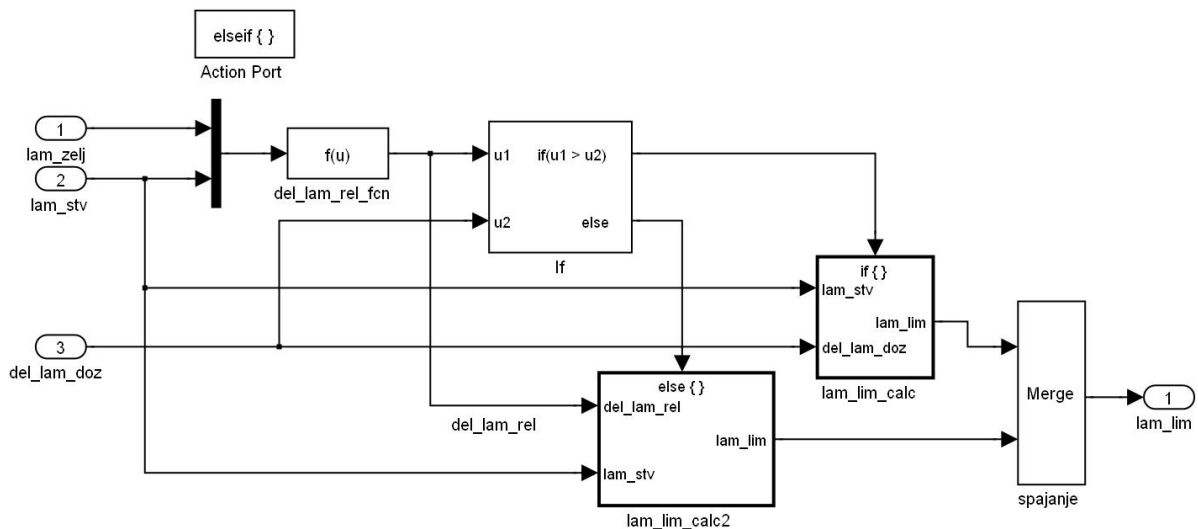
Sl. 10.9 Određivanje masenog protoka vazduha kroz usisne kanale, na osnovu (neispravnih ili ispravnih) signala sa senzora (modul „dmacyldt\_ECU“ sa Sl. 10.7)



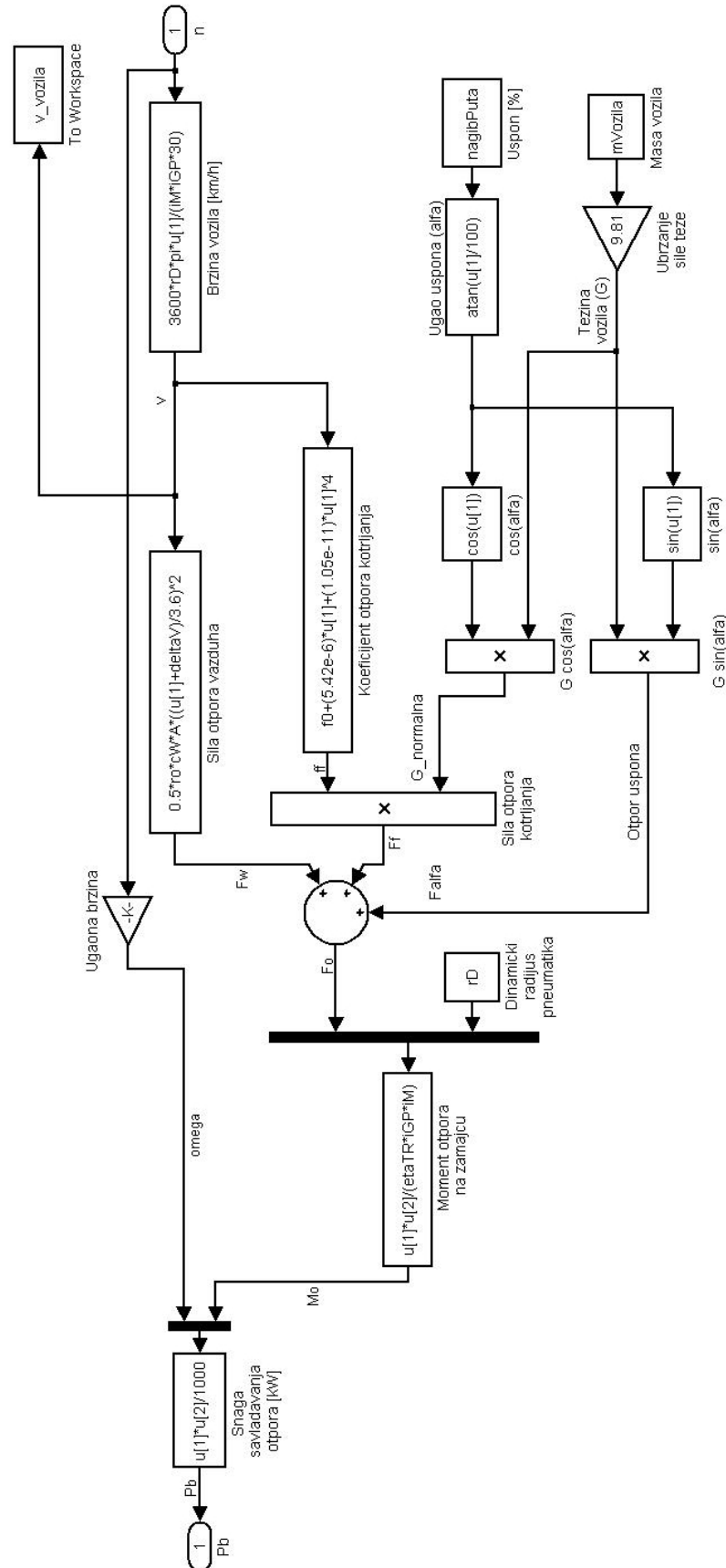
Sl. 10.10 Korigovanje potrebne količine goriva pri radu motora u povratnoj sprezi sa  $\lambda$ -sondom (Modul „Korekcija\_doze\_goriva“ sa Sl. 10.3)



Sl. 10.11 Korekcija koeficijenta viška vazduha na vrednost koju dozvoljava elektronska upravljačka jedinica (modul „lambda\_korekcija“ sa Sl. 10.10)



Sl. 10.12 Struktura modula „smesa\_bogatija“ i „smesa\_siromasnija“ sa Sl. 10.11



Sl. 10.13 Određivanje snage potrebne za savladavanje otpora kretanja vozila (modul „Pb“ sa Sl. 10.3)

## P2 Listing izvornog koda programa NSimulatorMenadzer i pratećih programa

### P2.1 NSimulatorMenadzer – izvorni kod

```

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
NSimulatorMenadzer.m

Ovim fajlom se generisu podaci koji se dobijaju simulacijom za slucajeve ustaljenog rada motora. Pokrece se
simulacija MVEM modela motora jedanput za slucaj kada nema greske, a onda jos po jednom za svaku gresku koja
se zeli simulirati. Konkretno, ovde bi to trebalo da se cini 14 puta (1+13, jednom za rad motora bez greske i
13 puta za simuliranje gresaka. Na kraju se podaci dobijeni simulacijama snimaju u dva "xls" fajla. Broj
podataka menja se izmenom koraka simulacije (simKorak), ali i duzinom trajanja cele simulacije (simKraj).
Ugao leptira regulise se parametrom "a", ali je izvršena transformacija ugla leptira u otvor leptira izrazen
u procentima, sto vise odgovara realnoj situaciji, jer TPS (Throttle Position Sensor) daje procentualno otvor
leptira. Inace, simulira se rad zagrejanog motora na nekim srednjim rezimima, kada EGR ventil treba da bude
potpuno otvoren, zbog minimizovanja emisije NOx.
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

clear;
postaviCodePage;
imaProtokomer=0;
% Pomocu koeficijenta "imaProtokomer" bira se da li motor radi po "speed-density" regulaciji ili ima
protokomer. Ako je imaProtokomer=1, onda se protok fals vazduha oduzima od protoka pored usisnih ventila pri
racunanju doze goriva, a ako je imaProtokomer=0, onda i fals vazduh ulazi u kalkulaciju potrebne doze
goriva.
kEGR=1;
% kEGR=1 znaci da EGR ventil nije zaglavljnjen i da je recirkulacija tolika, kolika bi i trebalo da bude -
regulacija u povratnoj sprezi, dok kEGR=0 znaci da se simulira rad motora kada nije potrebna EG
recirkulacija.
deltaLambdaDoz=1000;
% Ako je deltaLambdaDoz=0, onda nema lambda-korekcije tj. motor radi bez povratne sprege.
if deltaLambdaDoz>0
    uPovSprezi=1;
else
    uPovSprezi=0;
end

%{
Malo uputstvo za vrednosti parametara pri simulacijama rada motora u lambda-rezimu i bez lambda-rezima:
    U lambda-rezimu, nema EGR neispravnosti (protok EGR maksimalno moguci za taj rezim): uPovSprezi=1,
    uEGRrezimu=1, kEGR=1, deltaLambdaDoz<>0
    U lambda-rezimu, EGR zagusen u nekoj meri:
    uPovSprezi=1, uEGRrezimu=1, kEGR=neka vrednost,koja odgovara velicini zagusenja, deltaLambdaDoz<>0
    Nije u lambda-rezimu, nema EGR neispravnosti (EGR zatvoren):
    uPovSprezi=0, uEGRrezimu=0, kEGR=0, deltaLambdaDoz=0
    Nije u lambda-rezimu, EGR ventil treba da bude zatvoren, a nije (ne "dihtuje"):
    uPovSprezi=0, uEGRrezimu=0, kEGR=neka vrednost, deltaLambdaDoz=0
%}

h = waitbar(0,'Simulacija u toku. Sacekajte...'); % Generisanje vejtbara
ukupnoSimulacija = 11; % (1 inicijalna simulacija + 1 bez gresaka + 4 MAF + 3 FV + 2 EGR)
if imaProtokomer==0
    ukupnoSimulacija=ukupnoSimulacija+4; % (1 inicijalna + 1 bez gresaka + 4MAP + 4IAT + 3FV + 2EGR)
end
podaciOmotoru;
svrhaSimulacije=1;
% Ako je svrhaSimulacije=0, onda je svrha dobijanje podataka za obuku, a ako je svrhaSimulacije=1, onda je
svrha dobijanje podataka za testiranje.
simulinkFajl='NSimulator';
% definisanje koraka i vremena trajanja simulacije [s]
simKorak=0.25;
simKraj=25; % Za slucaj stacionarnog rada motora parametar "simKraj" mora da bude manji od vremena tl

% ovo je interval iz kojeg se uzimaju podaci za kasnije obucavanje neuronskih mreza, da bi se izbegli podaci
koji se dobijaju od momenta pocetka simulacije (kada vaze pocetni uslovi) pa do uspostavljanja stabilnog
stanja, jer ti podaci ne odgovaraju zadatom polozaju leptira
interval=1+(22/25)*simKraj/simKorak:simKraj/simKorak+1; % 13 uzoraka po simulaciji (3 sekunde x 4 uzorka + 1)
% koeficijenti suma za generisanje realnijih podataka sa sumom. Odredjeni su na osnovu oscilovanja vrednosti
eksperimentalnih podataka na stacionarnom rezimu iz VEMS-a
sumPrit=0.005;
sumTemp=0.0;
sumRevNum=0.005;
sumProtok=0.000025;
sumImpuls=0.00008;
sumLambda=0.01;
% Koeficijenti za generisanje pulsacija
odstupanjePrit=2; % procenat odstupanja pritiska u usisnom kolektoru +/- 2%

```



```

amplRevNum=10/1000; % amplituda odstupanja broja obrtaja je 10, ovde se pretvara u jedinice koje se koriste u
                    simulink modelu
odstupanjeProtoka=0.7; % procenat odstupanja svih protoka
opterećenje = 8; % Izbor otvora leptira u procentima
brojStanja = 14; % brojStanja = broj neispravnosti koje se prate + stanje bez neispravnosti
% Izbor stepena prenosa
%iM=iI;
%iM=iIII;
iM=iIII;
%iM=iIV;
%iM=iV;

switch iM
    case iI
        stText= 'I';
        stepenPrenosa = 1;
    case iIII
        stText='III';
        stepenPrenosa = 2;
    case iIII
        stText='III';
        stepenPrenosa = 3;
    case iIV
        stText='IV';
        stepenPrenosa = 4;
    otherwise
        stText='V';
        stepenPrenosa = 5;
end
% Pretvaranje procentualnog otvora leptira u ugao leptira (jer je to promenljiva koju trazi originalni
Hendriksov model)
a = procenatToStepen(opterećenje,alphaFull,alphaNula); % Inicijalizacija kvadratnog talasa
t1=30; % Trenutak naglog otvaranja leptira
t2=33; % Trenutak naglog zatvaranja leptira
t3=36; % Trenutak povratka leptira u pocetni položaj
tausquare=.05; % vremenska konstanta za kvadratni talas
square0=a; % amplituda kvadratnog talasa
square1=10; % amplituda kvadratnog talasa
squareini=a; % inicijalna amplituda
xStart=[0.23926 .9 a Ta]; % Ovo su pocetni uslovi za inicijalnu simulaciju
pi2 = xStart(1);
Ti2ini = xStart(4);
n2ini = xStart(2);
aini = a;
alpha = a;
pi2ini = pi2;

povrsinaOtvora=0; % nema nezelenog vazduha
maxEGR=0.2; % koeficijent maksimalne recirkulacije (ovo znaci da maksimalni maseni protok EGR iznosi
            20% od masenog protoka kroz usisni kolektor)

kSensMAFgain=1;
kSensMAFoffset=0; % Ovi parametri oznacavaju da je MAF senzor ispravan ili ga motor ne poseduje
kSensPritGain=1;
kSensPritOffset=0; % Ovi parametri oznacavaju da nema greske MAP senzora
kSensTempGain=1;
kSensTempOffset=0; % Ovi parametri oznacavaju da nema greske IAT senzora

% Inicijalna simulacija - postavljanje pocetnih uslova (Vrednosti parametara na kraju ove inicijalne simulacije
koriste se kao pocetne vrednosti za stvarne simulacije)
tekucaSimulacija = 1;
sim(simulinkFajl,[-8 -1]);
waitbar(tekucaSimulacija / ukupnoSimulacija); % Vejtbar prikazuje napredak simulacija
InitdataForNN=[tout xout(:,4) xout(:,2) xout(:,1) xout(:,3)];

pi2ini = xFinal(1);
Ti2ini = xFinal(2);
n2ini = xFinal(3);

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Izvodjenje simulacija motora %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% Simulacija rada motora bez gresaka

sim(simulinkFajl, [0 simKorak:simKorak:simKraj-simKorak simKraj]);
tekucaSimulacija = tekucaSimulacija +1;
waitbar(tekucaSimulacija / ukupnoSimulacija); % Vejtbar prikazuje napredak simulacija
vreme=tout(interval);
ugaoLeptira=Angle.signals.values(interval);
pritisak=generisiPulsacijeProc(InletPressure.signals.values(interval),odstupanjePrit);
temperatura=generisiSum(InletTemp.signals.values(interval),sumTemp);
temperaturaStvarna=temperatura; % Inicijalizuje se matrica sa stvarnom temperaturom (temperaturom koju očitava
ispravan IAT senzor).
revNum=CrankshaftSpeed.signals.values(interval);
dataRefined=[ugaoLeptira pritisak temperatura revNum];
dataNoFault=dataRefined;
dataForNN=[tout xout(:,4) xout(:,2) xout(:,1) xout(:,3)];
normalizedDataForNN=[tout (xout(:,4)-alphaNula)/(alphaFull-alphaNula)
(xout(:,2)-minTemperatura)/(maxTemperatura-minTemperatura)
(xout(:,1)-minPritisak)/(maxPritisak-minPritisak)

```

```

(xout(:,3)-minRPM)/(maxRPM-minRPM)];

% Protoci
protokThrottle=generisiPulsacijeProc(throttleFlow.signals.values(interval),
odstupanjeProtoka); % Protok pored leptira
protokValve=portFlow.signals.values(interval); % Protok kroz usisne kanale
protokRecirk=EGRFlow.signals.values(interval); % Umanjeni EGR protok usled prigusenja
protokFals=leakFlow.signals.values(interval); % Protok fals vazduha
protokRecirkNF=EGRNFFlow.signals.values(interval); % Puni EGR protok
% Vrednosti koeficijenta viska vazduha (lambda)
lambda_target=lambda_zeljeno.signals.values(interval); % Zeljena
lam_real=lam_stv.signals.values(interval); % Stvarna
lam_limit=lam_lim.signals.values(interval); % Maksimalno korigovana u slucaju velikog odstupanja
dmfdt_limit=dmfdt_lim.signals.values(interval); % Maseni protok goriva korigovan prema lam_limit
dmfdt_nekor=dmfdt_nekorig.signals.values(interval); % Maseni protok goriva izracunat prema zeljenoj vrednosti
lambda (lambda_target)

duzinaImpulsa=PW_sec.signals.values(interval); % Duzina impulsa ubrizgavanja [s]
v_voz=v_vozila.signals.values(interval); % Relativna brzina vazduha u odnosu na vozilo [km/h]
TiSenzorska=[];
protokKojiVidiECU=[];
dotokUCilindre=[];

if imaProtokomer==0
% Priprema za simulaciju greske senzora temperature - 15C niza vrednost (greska ofseta)
% Napomena: odstupanja vrednosti koje daje senzor temperature su u odnosu na Celzijusovu skalu
kSensTempOffset=-15;
sim(simulinkFajl, [0 simKorak:simKorak:simKraj-simKorak simKraj]);
tekucaSimulacija = tekucaSimulacija +1;
waitbar(tekucaSimulacija / ukupnoSimulacija); % Vejtbar prikazuje napredak simulacija
vreme=[vreme; tout(interval)];
ugaoLeptira=[ugaoLeptira; Angle.signals.values(interval)];
pritisak=[pritisak; InletPressure.signals.values(interval)];
tempLastSim=generisiSum(InletTemp.signals.values(interval),sumTemp);
temperatura=[temperatura; tempLastSim-(1-kSensTempGain)*(tempLastSim-273)+kSensTempOffset];
revNum=[revNum; CrankshaftSpeed.signals.values(interval)];
temperaturaStvarna=[temperaturaStvarna; tempLastSim];
protokThrottleLastSim=generisiPulsacijeProc(throttleFlow.signals.values(interval), odstupanjeProtoka);
protokThrottle=[protokThrottle;protokThrottleLastSim];
protokValve=[protokValve; portFlow.signals.values(interval)];
protokRecirk=[protokRecirk; EGRFlow.signals.values(interval)];
protokFals=[protokFals; leakFlow.signals.values(interval)];
protokRecirkNF=[protokRecirkNF; EGRNFFlow.signals.values(interval)];
lambda_target=[lambda_target; lambda_zeljeno.signals.values(interval)];
lam_real=[lam_real; lam_stv.signals.values(interval)];
lam_limit=[lam_limit; lam_lim.signals.values(interval)];
dmfdt_limit=[dmfdt_limit; dmfdt_lim.signals.values(interval)];
dmfdt_nekor=[dmfdt_nekor; dmfdt_nekorig.signals.values(interval)];
duzinaImpulsa=[duzinaImpulsa;PW_sec.signals.values(interval)]; % Trajanje ubrizgavanja [s]
v_voz=[v_voz; v_vozila.signals.values(interval)];
TiSenzorska=[TiSenzorska TiSenzor.signals.values(interval)];
protokKojiVidiECU=[protokKojiVidiECU protokUsisKorigovan.signals.values(interval)];
dotokUCilindre=[dotokUCilindre dmacyldt.signals.values(interval)];

% Priprema za simulaciju greske senzora temperature - 10C niza vrednost (greska ofseta)
kSensTempOffset=-10;
sim(simulinkFajl, [0 simKorak:simKorak:simKraj-simKorak simKraj]);
tekucaSimulacija = tekucaSimulacija +1;
waitbar(tekucaSimulacija / ukupnoSimulacija); % Vejtbar prikazuje napredak simulacija
vreme=[vreme; tout(interval)];
ugaoLeptira=[ugaoLeptira; Angle.signals.values(interval)];
pritisak=[pritisak; InletPressure.signals.values(interval)];
tempLastSim=generisiSum(InletTemp.signals.values(interval),sumTemp);
temperatura=[temperatura; tempLastSim-(1-kSensTempGain)*(tempLastSim-273)+kSensTempOffset];
revNum=[revNum; CrankshaftSpeed.signals.values(interval)];
temperaturaStvarna=[temperaturaStvarna; tempLastSim];
protokThrottleLastSim=generisiPulsacijeProc(throttleFlow.signals.values(interval), odstupanjeProtoka);
protokThrottle=[protokThrottle;protokThrottleLastSim];
protokValve=[protokValve; portFlow.signals.values(interval)];
protokRecirk=[protokRecirk; EGRFlow.signals.values(interval)];
protokFals=[protokFals; leakFlow.signals.values(interval)];
protokRecirkNF=[protokRecirkNF; EGRNFFlow.signals.values(interval)];
lambda_target=[lambda_target; lambda_zeljeno.signals.values(interval)];
lam_real=[lam_real; lam_stv.signals.values(interval)];
lam_limit=[lam_limit; lam_lim.signals.values(interval)];
dmfdt_limit=[dmfdt_limit; dmfdt_lim.signals.values(interval)];
dmfdt_nekor=[dmfdt_nekor; dmfdt_nekorig.signals.values(interval)];
duzinaImpulsa=[duzinaImpulsa; PW_sec.signals.values(interval)]; % Trajanje ubrizgavanja [s]
v_voz=[v_voz; v_vozila.signals.values(interval)];
TiSenzorska=[TiSenzorska; TiSenzor.signals.values(interval)];
protokKojiVidiECU=[protokKojiVidiECU; protokUsisKorigovan.signals.values(interval)];
dotokUCilindre=[dotokUCilindre; dmacyldt.signals.values(interval)];

% Priprema za simulaciju greske senzora temperature - 10C visa vrednost (greska ofseta)
kSensTempOffset=10;
sim(simulinkFajl, [0 simKorak:simKorak:simKraj-simKorak simKraj]);
tekucaSimulacija = tekucaSimulacija +1;
waitbar(tekucaSimulacija / ukupnoSimulacija); % Vejtbar prikazuje napredak simulacija
vreme=[vreme; tout(interval)];
ugaoLeptira=[ugaoLeptira; Angle.signals.values(interval)];
pritisak=[pritisak; InletPressure.signals.values(interval)];
tempLastSim=generisiSum(InletTemp.signals.values(interval),sumTemp);

```

```

temperatura=[temperatura; tempLastSim-(1-kSensTempGain)*(tempLastSim-273)+kSensTempOffset];
revNum=[revNum; CrankshaftSpeed.signals.values(interval)];
temperaturaStvarna=[temperaturaStvarna; tempLastSim];
protokThrottleLastSim=generisiPulsacijeProc(throttleFlow.signals.values(interval), odstupanjeProtoka);
protokThrottle=[protokThrottle;protokThrottleLastSim];
protokValve=[protokValve; portFlow.signals.values(interval)];
protokRecirk=[protokRecirk; EGRFlow.signals.values(interval)];
protokFals=[protokFals; leakFlow.signals.values(interval)];
protokRecirkNF=[protokRecirkNF; EGRNFFlow.signals.values(interval)];
lambda_target=[lambda_target; lambda_zeljeno.signals.values(interval)];
lam_real=[lam_real; lam_stv.signals.values(interval)];
lam_limit=[lam_limit; lam_lim.signals.values(interval)];
dmfdt_limit=[dmfdt_limit; dmfdt_lim.signals.values(interval)];
dmfdt_nekor=[dmfdt_nekor; dmfdt_nekorig.signals.values(interval)];
duzinaImpulsa=[duzinaImpulsa; PW_sec.signals.values(interval)]; % Trajanje ubrizgavanja [s]
v_voz=[v_voz; v_vozila.signals.values(interval)];
TiSenzorska=[TiSenzorska; TiSenzor.signals.values(interval)];
protokKojiVidiECU=[protokKojiVidiECU; protokUsisKorigovan.signals.values(interval)];
dotokUCilindre=[dotokUCilindre; dmacyldt.signals.values(interval)];

% Priprema za simulaciju greske senzora temperature - 15C visa vrednost (greska ofseta)
kSensTempOffset=15;
sim(simulinkFajl, [0 simKorak:simKorak:simKraj-simKorak simKraj]);
tekucaSimulacija = tekucaSimulacija +1;
waitbar(tekucaSimulacija / ukupnoSimulacija); % Vejtbar prikazuje napredak simulacija
vreme=[vreme; tout(interval)];
ugaoLeptira=[ugaoLeptira; Angle.signals.values(interval)];
pritisak=[pritisak; InletPressure.signals.values(interval)];
tempLastSim=generisiSum(InletTemp.signals.values(interval),sumTemp);
temperatura=[temperatura; tempLastSim-(1-kSensTempGain)*(tempLastSim-273)+kSensTempOffset];
revNum=[revNum; CrankshaftSpeed.signals.values(interval)];
temperaturaStvarna=[temperaturaStvarna; tempLastSim];
protokThrottleLastSim=generisiPulsacijeProc(throttleFlow.signals.values(interval), odstupanjeProtoka);
protokThrottle=[protokThrottle;protokThrottleLastSim];
protokValve=[protokValve; portFlow.signals.values(interval)];
protokRecirk=[protokRecirk; EGRFlow.signals.values(interval)];
protokFals=[protokFals; leakFlow.signals.values(interval)];
protokRecirkNF=[protokRecirkNF; EGRNFFlow.signals.values(interval)];
lambda_target=[lambda_target; lambda_zeljeno.signals.values(interval)];
lam_real=[lam_real; lam_stv.signals.values(interval)];
lam_limit=[lam_limit; lam_lim.signals.values(interval)];
dmfdt_limit=[dmfdt_limit; dmfdt_lim.signals.values(interval)];
dmfdt_nekor=[dmfdt_nekor; dmfdt_nekorig.signals.values(interval)];
duzinaImpulsa=[duzinaImpulsa; PW_sec.signals.values(interval)]; % Trajanje ubrizgavanja [s]
v_voz=[v_voz; v_vozila.signals.values(interval)];
TiSenzorska=[TiSenzorska; TiSenzor.signals.values(interval)];
protokKojiVidiECU=[protokKojiVidiECU; protokUsisKorigovan.signals.values(interval)];
dotokUCilindre=[dotokUCilindre; dmacyldt.signals.values(interval)];
kSensTempGain=1; % Senzor temperature u usisnom kolektoru vraća se u ispravno stanje
kSensTempOffset=0;

% Priprema za simulaciju greske MAP senzora (greska ofseta). U slučaju kada motor nema MAF senzor tj. radi po
speed-density regulaciji doziranja goriva, pokretace se simulacije modela iz Simulinka za svaku gresku MAP
senzora, jer ovde MAP senzor ima mnogo znacajnu ulogu nego kod motora sa MAF senzorom. Greske ce se
simulirati pomocu odgovarajucih koeficijenata, cije vrednosti se prenose u sam simulink model i tamo uticu
na doziranje goriva. Pocetni uslovi p2ini, T2ini i n2ini koji su dobijeni posle prvog postavljanja pocetnih
uslova i dalje ostaju na snazi

pritisak=generisiPulsacijeProc(pritisak,odstupanjePrit);
pritisakStvarni=pritisak; % Ovde se inicijalizuje matrica koja ce sadržati stvarne vrednosti pritiska, a
ne one koje očitava neispravni MAP senzor
% Simulacija sa greskom MAP senzora - 10kPa niza vrednost
kSensPritGain=1;
kSensPritOffset=-0.1; % Koeficijent za simulaciju greske ofseta MAP senzora. Ovaj koeficijent ulazi u
simulaciju modela i utice na doziranje goriva.
sim(simulinkFajl, [0 simKorak:simKorak:simKraj-simKorak simKraj]);
tekucaSimulacija = tekucaSimulacija +1;
waitbar(tekucaSimulacija / ukupnoSimulacija); % Vejtbar prikazuje napredak simulacija
vreme=[vreme; tout(interval)];
ugaoLeptira=[ugaoLeptira; Angle.signals.values(interval)];
% Sada se vrsi transformacija stvarne vrednosti pritiska, da bi se dobila vrednost koju daje MAP senzor. Ovom
vrednoscu ce raspolagati ECU, pa je sa stanovista dijagnostike ona ovde vaznija.
pritisakLastSim=generisiPulsacijeProc(InletPressure.signals.values(interval),odstupanjePrit);
pritisak=[pritisak; kSensPritGain*pritisakLastSim+kSensPritOffset];
tempLastSim=generisiSum(InletTemp.signals.values(interval),sumTemp);
temperatura=[temperatura; tempLastSim];
revNum=[revNum; CrankshaftSpeed.signals.values(interval)];
temperaturaStvarna=[temperaturaStvarna; tempLastSim];
pritisakStvarni=[pritisakStvarni; pritisakLastSim];
protokThrottleLastSim=generisiPulsacijeProc(throttleFlow.signals.values(interval), odstupanjeProtoka);
protokThrottle=[protokThrottle;protokThrottleLastSim];
protokValve=[protokValve; portFlow.signals.values(interval)];
protokRecirk=[protokRecirk; EGRFlow.signals.values(interval)];
protokFals=[protokFals; leakFlow.signals.values(interval)];
protokRecirkNF=[protokRecirkNF; EGRNFFlow.signals.values(interval)];
lambda_target=[lambda_target; lambda_zeljeno.signals.values(interval)];
lam_real=[lam_real; lam_stv.signals.values(interval)];
lam_limit=[lam_limit; lam_lim.signals.values(interval)];
dmfdt_limit=[dmfdt_limit; dmfdt_lim.signals.values(interval)];
dmfdt_nekor=[dmfdt_nekor; dmfdt_nekorig.signals.values(interval)];

```

```

duzinaImpulsa=[duzinaImpulsa; PW_sec.signals.values(interval)]; % Trajanje ubrizgavanja [s]
v_voz=[v_voz; v_vozila.signals.values(interval)];

% Simulacija greske MAP senzora - 5kPa niza vrednost (greska ofseta). Pocetni uslovi pi2ini, T2ini i n2ini,
dobijeni posle prvog postavljanja pocetnih uslova, ostaju na snazi
kSensPritOffset=-0.05; % Simulacija sa greskom MAP senzora - 5kPa niza vrednost
sim(simulinkFajl, [0 simKorak:simKorak:simKraj-simKorak simKraj]);
tekucaSimulacija = tekucaSimulacija +1;
waitbar(tekucaSimulacija / ukupnoSimulacija); % Vejtbar prikazuje napredak simulacija
vreme=[vreme; tout(interval)];
ugaoLeptira=[ugaoLeptira; Angle.signals.values(interval)];
% Sada se vrsi transformacija stvarne vrednosti pritiska, da bi se dobila vrednost koju daje MAP senzor. Ovom
vrednoscu ce raspolagati ECU, pa je sa stanovista dijagnostike ona ovde vaznija.
pritisakLastSim=generisiPulsacijeProc(InletPressure.signals.values(interval),odstupanjePrit);
pritisak=[pritisak; kSensPritGain*pritisakLastSim+kSensPritOffset];
tempLastSim=generisiSum(InletTemp.signals.values(interval),sumTemp);
temperatura=[temperatura; tempLastSim];
revNum=[revNum; CrankshaftSpeed.signals.values(interval)];
temperaturaStvarna=[temperaturaStvarna; tempLastSim];
pritisakStvarni=[pritisakStvarni; pritisakLastSim];
protokThrottleLastSim=generisiPulsacijeProc(throttleFlow.signals.values(interval), odstupanjeProtoka);
protokThrottle=[protokThrottle;protokThrottleLastSim];
protokValve=[protokValve; portFlow.signals.values(interval)];
protokRecirk=[protokRecirk; EGRFlow.signals.values(interval)];
protokFals=[protokFals; leakFlow.signals.values(interval)];
protokRecirkNF=[protokRecirkNF; EGRNFFlow.signals.values(interval)];
lambda_target=[lambda_target; lambda_zeljeno.signals.values(interval)];
lam_real=[lam_real; lam_stv.signals.values(interval)];
lam_limit=[lam_limit; lam_lim.signals.values(interval)];
dmfdt_limit=[dmfdt_limit; dmfdt_lim.signals.values(interval)];
dmfdt_nekor=[dmfdt_nekor; dmfdt_nekorig.signals.values(interval)];
duzinaImpulsa=[duzinaImpulsa; PW_sec.signals.values(interval)]; % Trajanje ubrizgavanja [s]
v_voz=[v_voz; v_vozila.signals.values(interval)];

% Simulacija greske MAP senzora - 5kPa visa vrednost (greska ofseta). Pocetni uslovi pi2ini, T2ini i n2ini,
dobijeni posle prvog postavljanja pocetnih uslova, ostaju na snazi
kSensPritOffset=0.05; % Simulacija sa greskom MAP senzora - 5kPa visa vrednost
sim(simulinkFajl, [0 simKorak:simKorak:simKraj-simKorak simKraj]);
tekucaSimulacija = tekucaSimulacija +1;
waitbar(tekucaSimulacija / ukupnoSimulacija); % Vejtbar prikazuje napredak simulacija
vreme=[vreme; tout(interval)];
ugaoLeptira=[ugaoLeptira; Angle.signals.values(interval)];
% Sada se vrsi transformacija stvarne vrednosti pritiska, da bi se dobila vrednost koju daje MAP senzor. Ovom
vrednoscu ce raspolagati ECU, pa je sa stanovista dijagnostike ona ovde vaznija.
pritisakLastSim=generisiPulsacijeProc(InletPressure.signals.values(interval),odstupanjePrit);
pritisak=[pritisak; kSensPritGain*pritisakLastSim+kSensPritOffset];
tempLastSim=generisiSum(InletTemp.signals.values(interval),sumTemp);
temperatura=[temperatura; tempLastSim];
revNum=[revNum; CrankshaftSpeed.signals.values(interval)];
temperaturaStvarna=[temperaturaStvarna; tempLastSim];
pritisakStvarni=[pritisakStvarni; pritisakLastSim];
protokThrottleLastSim=generisiPulsacijeProc(throttleFlow.signals.values(interval), odstupanjeProtoka);
protokThrottle=[protokThrottle;protokThrottleLastSim];
protokValve=[protokValve; portFlow.signals.values(interval)];
protokRecirk=[protokRecirk; EGRFlow.signals.values(interval)];
protokFals=[protokFals; leakFlow.signals.values(interval)];
protokRecirkNF=[protokRecirkNF; EGRNFFlow.signals.values(interval)];
lambda_target=[lambda_target; lambda_zeljeno.signals.values(interval)];
lam_real=[lam_real; lam_stv.signals.values(interval)];
lam_limit=[lam_limit; lam_lim.signals.values(interval)];
dmfdt_limit=[dmfdt_limit; dmfdt_lim.signals.values(interval)];
dmfdt_nekor=[dmfdt_nekor; dmfdt_nekorig.signals.values(interval)];
duzinaImpulsa=[duzinaImpulsa; PW_sec.signals.values(interval)]; % Trajanje ubrizgavanja [s]
v_voz=[v_voz; v_vozila.signals.values(interval)];

% Simulacija greske MAP senzora - 10kPa visa vrednost (greska ofseta). Pocetni uslovi pi2ini, T2ini i n2ini,
dobijeni posle prvog postavljanja pocetnih uslova, ostaju na snazi
kSensPritOffset=0.1; % Simulacija sa greskom MAP senzora - 10% visa vrednost
sim(simulinkFajl, [0 simKorak:simKorak:simKraj-simKorak simKraj]);
tekucaSimulacija = tekucaSimulacija +1;
waitbar(tekucaSimulacija / ukupnoSimulacija); % Vejtbar prikazuje napredak simulacija
vreme=[vreme; tout(interval)];
ugaoLeptira=[ugaoLeptira; Angle.signals.values(interval)];
% Sada se vrsi transformacija stvarne vrednosti pritiska, da bi se dobila vrednost koju daje MAP senzor. Ovom
vrednoscu ce raspolagati ECU, pa je sa stanovista dijagnostike ona ovde vaznija.
pritisakLastSim=generisiPulsacijeProc(InletPressure.signals.values(interval),odstupanjePrit);
pritisak=[pritisak; kSensPritGain*pritisakLastSim+kSensPritOffset];
tempLastSim=generisiSum(InletTemp.signals.values(interval),sumTemp);
temperatura=[temperatura; tempLastSim];
revNum=[revNum; CrankshaftSpeed.signals.values(interval)];
temperaturaStvarna=[temperaturaStvarna; tempLastSim];
pritisakStvarni=[pritisakStvarni; pritisakLastSim];
protokThrottleLastSim=generisiPulsacijeProc(throttleFlow.signals.values(interval), odstupanjeProtoka);
protokThrottle=[protokThrottle;protokThrottleLastSim];
protokValve=[protokValve; portFlow.signals.values(interval)];
protokRecirk=[protokRecirk; EGRFlow.signals.values(interval)];
protokFals=[protokFals; leakFlow.signals.values(interval)];
protokRecirkNF=[protokRecirkNF; EGRNFFlow.signals.values(interval)];
lambda_target=[lambda_target; lambda_zeljeno.signals.values(interval)];
lam_real=[lam_real; lam_stv.signals.values(interval)];

```

```

lam_limit=[lam_limit; lam_lim.signals.values(interval)];
dmfdt_limit=[dmfdt_limit; dmfdt_lim.signals.values(interval)];
dmfdt_nekor=[dmfdt_nekor; dmfdt_nekorig.signals.values(interval)];
duzinaImpulsa=[duzinaImpulsa; PW_sec.signals.values(interval)]; % Trajanje ubrizgavanja [s]
v_voz=[v_voz; v_vozila.signals.values(interval)];
kSensPritOffset=0;
end

if imaProtokomer==1
% Simulacija greske senzora pritiska - 10 kPa niza vrednost (greska ofseta). U slucaju kada motor ima maseni
protokomer nece se pokretati simulacije modela iz Simulinka, nego ce se greske ofseta MAP senzora simulirati
primenom odgovarajucih koeficijenata na vrednosti pritiska, dobijene pri simulaciji rada motora bez gresaka
kSensTempGain=1; % senzor temperature se postavlja u ispravno stanje
kSensPritGain=1;
kSensPritOffset=-0.1; % 0.1 bar offset nanize
pritisakStvarni=pritisak;
vreme=[vreme; tout(interval)];
ugaoLeptira=[ugaoLeptira; Angle.signals.values(interval)];
pritisakLastSim=generisiPulsacijeProc(InletPressure.signals.values(interval),odstupanjePrit);
pritisak=[pritisak; kSensPritGain*pritisakLastSim+kSensPritOffset];
tempLastSim=generisiSum(InletTemp.signals.values(interval),sumTemp);
temperatura=[temperatura; tempLastSim];
revNum=[revNum; CrankshaftSpeed.signals.values(interval)];
temperaturaStvarna=[temperaturaStvarna; tempLastSim];
pritisakStvarni=[pritisakStvarni; pritisakLastSim];
protokThrottleLastSim=generisiPulsacijeProc(throttleFlow.signals.values(interval), odstupanjeProtoka);
protokThrottle=[protokThrottle;protokThrottleLastSim];
protokValve=[protokValve; portFlow.signals.values(interval)];
protokRecirk=[protokRecirk; EGRFlow.signals.values(interval)];
protokFals=[protokFals; leakFlow.signals.values(interval)];
protokRecirkNF=[protokRecirkNF; EGRNFFlow.signals.values(interval)];
lambda_target=[lambda_target; lambda_zeljeno.signals.values(interval)];
lam_real=[lam_real; lam_stv.signals.values(interval)];
lam_limit=[lam_limit; lam_lim.signals.values(interval)];
dmfdt_limit=[dmfdt_limit; dmfdt_lim.signals.values(interval)];
dmfdt_nekor=[dmfdt_nekor; dmfdt_nekorig.signals.values(interval)];
duzinaImpulsa=[duzinaImpulsa; PW_sec.signals.values(interval)]; % Trajanje ubrizgavanja [s]
v_voz=[v_voz; v_vozila.signals.values(interval)];

% Simulacija greske senzora pritiska - 5 kPa niza vrednost (greska ofseta)
kSensPritGain=1;
kSensPritOffset=-0.05; % 0.05 bar ofset nainize
vreme=[vreme; tout(interval)];
ugaoLeptira=[ugaoLeptira; Angle.signals.values(interval)];
pritisakLastSim=generisiPulsacijeProc(InletPressure.signals.values(interval),odstupanjePrit);
pritisak=[pritisak; kSensPritGain*pritisakLastSim+kSensPritOffset];
tempLastSim=generisiSum(InletTemp.signals.values(interval),sumTemp);
temperatura=[temperatura; tempLastSim];
revNum=[revNum; CrankshaftSpeed.signals.values(interval)];
temperaturaStvarna=[temperaturaStvarna; tempLastSim];
pritisakStvarni=[pritisakStvarni; pritisakLastSim];
protokThrottleLastSim=generisiPulsacijeProc(throttleFlow.signals.values(interval), odstupanjeProtoka);
protokThrottle=[protokThrottle;protokThrottleLastSim];
protokValve=[protokValve; portFlow.signals.values(interval)];
protokRecirk=[protokRecirk; EGRFlow.signals.values(interval)];
protokFals=[protokFals; leakFlow.signals.values(interval)];
protokRecirkNF=[protokRecirkNF; EGRNFFlow.signals.values(interval)];
lambda_target=[lambda_target; lambda_zeljeno.signals.values(interval)];
lam_real=[lam_real; lam_stv.signals.values(interval)];
lam_limit=[lam_limit; lam_lim.signals.values(interval)];
dmfdt_limit=[dmfdt_limit; dmfdt_lim.signals.values(interval)];
dmfdt_nekor=[dmfdt_nekor; dmfdt_nekorig.signals.values(interval)];
duzinaImpulsa=[duzinaImpulsa; PW_sec.signals.values(interval)]; % Trajanje ubrizgavanja [s]
v_voz=[v_voz; v_vozila.signals.values(interval)];

% Simulacija greske senzora pritiska - 5 kPa visa vrednost (greska ofseta)
kSensPritGain=1;
kSensPritOffset=0.05; % 0.05 bar ofset navise
vreme=[vreme; tout(interval)];
ugaoLeptira=[ugaoLeptira; Angle.signals.values(interval)];
pritisakLastSim=generisiPulsacijeProc(InletPressure.signals.values(interval),odstupanjePrit);
pritisak=[pritisak; kSensPritGain*pritisakLastSim+kSensPritOffset];
tempLastSim=generisiSum(InletTemp.signals.values(interval),sumTemp);
temperatura=[temperatura; tempLastSim];
revNum=[revNum; CrankshaftSpeed.signals.values(interval)];
temperaturaStvarna=[temperaturaStvarna; tempLastSim];
pritisakStvarni=[pritisakStvarni; pritisakLastSim];
protokThrottleLastSim=generisiPulsacijeProc(throttleFlow.signals.values(interval), odstupanjeProtoka);
protokThrottle=[protokThrottle;protokThrottleLastSim];
protokValve=[protokValve; portFlow.signals.values(interval)];
protokRecirk=[protokRecirk; EGRFlow.signals.values(interval)];
protokFals=[protokFals; leakFlow.signals.values(interval)];
protokRecirkNF=[protokRecirkNF; EGRNFFlow.signals.values(interval)];
lambda_target=[lambda_target; lambda_zeljeno.signals.values(interval)];
lam_real=[lam_real; lam_stv.signals.values(interval)];
lam_limit=[lam_limit; lam_lim.signals.values(interval)];
dmfdt_limit=[dmfdt_limit; dmfdt_lim.signals.values(interval)];
dmfdt_nekor=[dmfdt_nekor; dmfdt_nekorig.signals.values(interval)];
duzinaImpulsa=[duzinaImpulsa; PW_sec.signals.values(interval)]; % Trajanje ubrizgavanja [s]
v_voz=[v_voz; v_vozila.signals.values(interval)];

```

```

% Simulacija greske senzora pritiska - 10 kPa visa vrednost (greska ofseta)
kSensPritGain=1;
kSensPritOffset=0.1; % 0.1 bar offset navise
vreme=[vreme; tout(interval)];
ugaoLeptira=[ugaoLeptira; Angle.signals.values(interval)];
pritisakLastSim=generisiPulsacijeProc(InletPressure.signals.values(interval),odstupanjePrit);
pritisak=[pritisak; kSensPritGain*pritisakLastSim+kSensPritOffset];
tempLastSim=generisiSum(InletTemp.signals.values(interval),sumTemp);
temperatura=[temperatura; tempLastSim];
revNum=[revNum; CrankshaftSpeed.signals.values(interval)];
temperaturaStvarna=[temperaturaStvarna; tempLastSim];
pritisakStvarni=[pritisakStvarni; pritisakLastSim];
protokThrottleLastSim=generisiPulsacijeProc(throttleFlow.signals.values(interval), odstupanjeProtoka);
protokThrottle=[protokThrottle;protokThrottleLastSim];
protokValve=[protokValve; portFlow.signals.values(interval)];
protokRecirk=[protokRecirk; EGRFlow.signals.values(interval)];
protokFals=[protokFals; leakFlow.signals.values(interval)];
protokRecirkNF=[protokRecirkNF; EGRNFFlow.signals.values(interval)];
% Uvodi se novi niz "protokThrottleFaulty", sa vrednostima signala neispravnog MAP senzora
protokThrottleFaulty=protokThrottle;
lambda_target=[lambda_target; lambda_zeljeno.signals.values(interval)];
lam_real=[lam_real; lam_stv.signals.values(interval)];
lam_limit=[lam_limit; lam_lim.signals.values(interval)];
dmfdt_limit=[dmfdt_limit; dmfdt_lim.signals.values(interval)];
dmfdt_nekor=[dmfdt_nekor; dmfdt_nekorig.signals.values(interval)];
duzinaImpulsa=[duzinaImpulsa; PW_sec.signals.values(interval)]; % Trajanje ubrizgavanja [s]
v_voz=[v_voz; v_vozila.signals.values(interval)];
kSensPritOffset=0; % MAP senzor se vraća u ispravno stanje

% Simulacija greske MAF senzora - 10% niza vrednost. Pocetni uslovi pi2ini, T2ini i n2ini, dobijeni posle prvog
postavljanja pocetnih uslova, ostaju na snazi
kSensMAFGain=0.90; % Simulacija sa greskom MAF senzora - 10% niza vrednost
sim(simulinkFajl, [0 simKorak:simKorak:simKraj-simKorak simKraj]);
tekucaSimulacija = tekucaSimulacija +1;
waitbar(tekucaSimulacija / ukupnoSimulacija); % Vejtbar prikazuje napredak simulacija
vreme=[vreme; tout(interval)];
ugaoLeptira=[ugaoLeptira; Angle.signals.values(interval)];
pritisakLastSim=generisiPulsacijeProc(InletPressure.signals.values(interval),odstupanjePrit);
pritisak=[pritisak; pritisakLastSim];
tempLastSim=generisiSum(InletTemp.signals.values(interval),sumTemp);
temperatura=[temperatura; tempLastSim];
revNum=[revNum; CrankshaftSpeed.signals.values(interval)];
temperaturaStvarna=[temperaturaStvarna; tempLastSim];
pritisakStvarni=[pritisakStvarni; pritisakLastSim];
protokThrottleLastSim=generisiPulsacijeProc(throttleFlow.signals.values(interval), odstupanjeProtoka);
protokThrottle=[protokThrottle;protokThrottleLastSim];
protokValve=[protokValve; portFlow.signals.values(interval)];
protokRecirk=[protokRecirk; EGRFlow.signals.values(interval)];
protokFals=[protokFals; leakFlow.signals.values(interval)];
protokRecirkNF=[protokRecirkNF; EGRNFFlow.signals.values(interval)];
protokThrottleFaulty=[protokThrottleFaulty; kSensMAFGain*protokThrottleLastSim];
lambda_target=[lambda_target; lambda_zeljeno.signals.values(interval)];
lam_real=[lam_real; lam_stv.signals.values(interval)];
lam_limit=[lam_limit; lam_lim.signals.values(interval)];
dmfdt_limit=[dmfdt_limit; dmfdt_lim.signals.values(interval)];
dmfdt_nekor=[dmfdt_nekor; dmfdt_nekorig.signals.values(interval)];
duzinaImpulsa=[duzinaImpulsa; PW_sec.signals.values(interval)]; % Trajanje ubrizgavanja [s]
v_voz=[v_voz; v_vozila.signals.values(interval)];

% Simulacija greske MAF senzora - 5% niza vrednost. Pocetni uslovi pi2ini, T2ini i n2ini koji su dobijeni posle
prvog postavljanja pocetnih uslova i dalje ostaju na snazi
kSensMAFGain=0.95; % Simulacija sa greskom MAF senzora - 5% niza vrednost
sim(simulinkFajl, [0 simKorak:simKorak:simKraj-simKorak simKraj]);
tekucaSimulacija = tekucaSimulacija +1;
waitbar(tekucaSimulacija / ukupnoSimulacija); % Vejtbar prikazuje napredak simulacija
vreme=[vreme; tout(interval)];
ugaoLeptira=[ugaoLeptira; Angle.signals.values(interval)];
pritisakLastSim=generisiPulsacijeProc(InletPressure.signals.values(interval),odstupanjePrit);
pritisak=[pritisak; pritisakLastSim];
tempLastSim=generisiSum(InletTemp.signals.values(interval),sumTemp);
temperatura=[temperatura; tempLastSim];
revNum=[revNum; CrankshaftSpeed.signals.values(interval)];
temperaturaStvarna=[temperaturaStvarna; tempLastSim];
pritisakStvarni=[pritisakStvarni; pritisakLastSim];
protokThrottleLastSim=generisiPulsacijeProc(throttleFlow.signals.values(interval), odstupanjeProtoka);
protokThrottle=[protokThrottle;protokThrottleLastSim];
protokValve=[protokValve; portFlow.signals.values(interval)];
protokRecirk=[protokRecirk; EGRFlow.signals.values(interval)];
protokFals=[protokFals; leakFlow.signals.values(interval)];
protokRecirkNF=[protokRecirkNF; EGRNFFlow.signals.values(interval)];
protokThrottleFaulty=[protokThrottleFaulty; kSensMAFGain*protokThrottleLastSim];
lambda_target=[lambda_target; lambda_zeljeno.signals.values(interval)];
lam_real=[lam_real; lam_stv.signals.values(interval)];
lam_limit=[lam_limit; lam_lim.signals.values(interval)];
dmfdt_limit=[dmfdt_limit; dmfdt_lim.signals.values(interval)];
dmfdt_nekor=[dmfdt_nekor; dmfdt_nekorig.signals.values(interval)];
duzinaImpulsa=[duzinaImpulsa; PW_sec.signals.values(interval)]; % Trajanje ubrizgavanja [s]
v_voz=[v_voz; v_vozila.signals.values(interval)];

```

```

% Simulacija greske MAF senzora - 5% visa vrednost. Pocetni uslovi pi2ini, T2ini i n2ini, koji su dobijeni
posle prvog postavljanja pocetnih uslova, i dalje ostaju na snazi
kSensMAFGain=1.05; % Simulacija sa greskom MAF senzora - 5% visa vrednost
sim(simulinkFajl, [0 simKorak:simKorak:simKraj-simKorak simKraj]);
tekucaSimulacija = tekucaSimulacija +1;
waitbar(tekucaSimulacija / ukupnoSimulacija); % Vejtbar prikazuje napredak simulacija
vreme=[vreme; tout(interval)];
ugaoLeptira=[ugaoLeptira; Angle.signals.values(interval)];
pritisakLastSim=generisiPulsacijeProc(InletPressure.signals.values(interval),odstupanjePrit);
pritisak=[pritisak; pritisakLastSim];
tempLastSim=generisiSum(InletTemp.signals.values(interval),sumTemp);
temperatura=[temperatura; tempLastSim];
revNum=[revNum; CrankshaftSpeed.signals.values(interval)];
temperaturaStvarna=[temperaturaStvarna; tempLastSim];
pritisakStvarni=[pritisakStvarni; pritisakLastSim];
protokThrottleLastSim=generisiPulsacijeProc(throttleFlow.signals.values(interval), odstupanjeProtoka);
protokThrottle=[protokThrottle;protokThrottleLastSim];
protokValve=[protokValve; portFlow.signals.values(interval)];
protokRecirk=[protokRecirk; EGRFlow.signals.values(interval)];
protokFals=[protokFals; leakFlow.signals.values(interval)];
protokRecirkNF=[protokRecirkNF; EGRNFFlow.signals.values(interval)];
protokThrottleFaulty=[protokThrottleFaulty; kSensMAFGain*protokThrottleLastSim];
lambda_target=[lambda_target; lambda_zeljeno.signals.values(interval)];
lam_real=[lam_real; lam_stv.signals.values(interval)];
lam_limit=[lam_limit; lam_lim.signals.values(interval)];
dmfdt_limit=[dmfdt_limit; dmfdt_lim.signals.values(interval)];
dmfdt_nekor=[dmfdt_nekor; dmfdt_nekorig.signals.values(interval)];
duzinaImpulsa=[duzinaImpulsa; PW_sec.signals.values(interval)]; % Trajanje ubrizgavanja [s]
v_voz=[v_voz; v_vozila.signals.values(interval)];

% Simulacija greske MAF senzora - 10% visa vrednost. Pocetni uslovi pi2ini, T2ini i n2ini, koji su dobijeni
posle prvog postavljanja pocetnih uslova, i dalje ostaju na snazi
kSensMAFGain=1.1; % Simulacija sa greskom MAF senzora - 10% visa vrednost
sim(simulinkFajl, [0 simKorak:simKorak:simKraj-simKorak simKraj]);
tekucaSimulacija = tekucaSimulacija +1;
waitbar(tekucaSimulacija / ukupnoSimulacija); % Vejtbar prikazuje napredak simulacija
vreme=[vreme; tout(interval)];
ugaoLeptira=[ugaoLeptira; Angle.signals.values(interval)];
pritisakLastSim=generisiPulsacijeProc(InletPressure.signals.values(interval),odstupanjePrit);
pritisak=[pritisak; pritisakLastSim];
tempLastSim=generisiSum(InletTemp.signals.values(interval),sumTemp);
temperatura=[temperatura; tempLastSim];
revNum=[revNum; CrankshaftSpeed.signals.values(interval)];
temperaturaStvarna=[temperaturaStvarna; tempLastSim];
pritisakStvarni=[pritisakStvarni; pritisakLastSim];
protokThrottleLastSim=generisiPulsacijeProc(throttleFlow.signals.values(interval), odstupanjeProtoka);
protokThrottle=[protokThrottle;protokThrottleLastSim];
protokValve=[protokValve; portFlow.signals.values(interval)];
protokRecirk=[protokRecirk; EGRFlow.signals.values(interval)];
protokFals=[protokFals; leakFlow.signals.values(interval)];
protokRecirkNF=[protokRecirkNF; EGRNFFlow.signals.values(interval)];
protokThrottleFaulty=[protokThrottleFaulty; kSensMAFGain*protokThrottleLastSim];
lambda_target=[lambda_target; lambda_zeljeno.signals.values(interval)];
lam_real=[lam_real; lam_stv.signals.values(interval)];
lam_limit=[lam_limit; lam_lim.signals.values(interval)];
dmfdt_limit=[dmfdt_limit; dmfdt_lim.signals.values(interval)];
dmfdt_nekor=[dmfdt_nekor; dmfdt_nekorig.signals.values(interval)];
duzinaImpulsa=[duzinaImpulsa; PW_sec.signals.values(interval)]; % Trajanje ubrizgavanja [s]
v_voz=[v_voz; v_vozila.signals.values(interval)];
kSensMAFGain=1; % MAF senzor se vraca u ispravno stanje
end

% Simulacija greske "fals vazduh 2 mm". Pocetni uslovi pi2ini, T2ini i n2ini, koji su dobijeni posle prvog
postavljanja pocetnih uslova, i dalje ostaju na snazi
dOtvora=0.002; % precnik otvora u metrima
povrsinaOtvora=dOtvora^2*3.14/4;
sim(simulinkFajl, [0 simKorak:simKorak:simKraj-simKorak simKraj]);
tekucaSimulacija = tekucaSimulacija+1;
waitbar(tekucaSimulacija / ukupnoSimulacija); % Vejtbar prikazuje napredak simulacija
vreme=[vreme; tout(interval)];
ugaoLeptira=[ugaoLeptira; Angle.signals.values(interval)];
pritisakLastSim=generisiPulsacijeProc(InletPressure.signals.values(interval),odstupanjePrit);
pritisak=[pritisak; pritisakLastSim];
tempLastSim=generisiSum(InletTemp.signals.values(interval),sumTemp);
temperatura=[temperatura; tempLastSim];
revNum=[revNum; CrankshaftSpeed.signals.values(interval)];
temperaturaStvarna=[temperaturaStvarna; tempLastSim];
pritisakStvarni=[pritisakStvarni; pritisakLastSim];
protokThrottleLastSim=generisiPulsacijeProc(throttleFlow.signals.values(interval), odstupanjeProtoka);
protokThrottle=[protokThrottle;protokThrottleLastSim];
protokValve=[protokValve; portFlow.signals.values(interval)];
protokRecirk=[protokRecirk; EGRFlow.signals.values(interval)];
protokFals=[protokFals; generisiPulsacijeProc(leakFlow.signals.values(interval),odstupanjeProtoka)];
protokRecirkNF=[protokRecirkNF; EGRNFFlow.signals.values(interval)];
if imaProtokomer==1
    protokThrottleFaulty=[protokThrottleFaulty; protokThrottleLastSim];
end
lambda_target=[lambda_target; lambda_zeljeno.signals.values(interval)];
lam_real=[lam_real; lam_stv.signals.values(interval)];
lam_limit=[lam_limit; lam_lim.signals.values(interval)];

```

```

dmfdt_limit=[dmfdt_limit; dmfdt_lim.signals.values(interval)];
dmfdt_nekor=[dmfdt_nekor; dmfdt_nekorig.signals.values(interval)];
duzinaImpulsa=[duzinaImpulsa; PW_sec.signals.values(interval)]; % Trajanje ubrizgavanja [s]
v_voz=[v_voz; v_vozila.signals.values(interval)];

% Simulacija greske "fals vazduh 4 mm". Pocetni uslovi pi2ini, T2ini i n2ini, koji su dobijeni posle prvog
% postavljanja pocetnih uslova, i dalje ostaju na snazi
dOtvora=0.004; % precnik otvora u metrima
povrsinaOtvora=dOtvora^2*3.14/4;
sim(simulinkFajl, [0 simKorak:simKorak:simKraj-simKorak simKraj]);
tekucaSimulacija = tekucaSimulacija +1;
waitbar(tekucaSimulacija / ukupnoSimulacija); % Vejtbar prikazuje napredak simulacija
vreme=[vreme; tout(interval)];
ugaoLeptira=[ugaoLeptira; Angle.signals.values(interval)];
pritisakLastSim=generisiPulsacijeProc(InletPressure.signals.values(interval),odstupanjePrit);
pritisak=[pritisak; pritisakLastSim];
tempLastSim=generisiSum(InletTemp.signals.values(interval),sumTemp);
temperatura=[temperatura; tempLastSim];
revNum=[revNum; CrankshaftSpeed.signals.values(interval)];
temperaturaStvarna=[temperaturaStvarna; tempLastSim];
pritisakStvarni=[pritisakStvarni; pritisakLastSim];
protokThrottleLastSim=generisiPulsacijeProc(throttleFlow.signals.values(interval), odstupanjeProtoka);
protokThrottle=[protokThrottle;protokThrottleLastSim];
protokValve=[protokValve; portFlow.signals.values(interval)];
protokRecirk=[protokRecirk; EGRFlow.signals.values(interval)];
protokFals=[protokFals; generisiPulsacijeProc(leakFlow.signals.values(interval),odstupanjeProtoka)];
protokRecirkNF=[protokRecirkNF; EGRNFFlow.signals.values(interval)];
lambda_target=[lambda_target; lambda_zeljeno.signals.values(interval)];
lam_real=[lam_real; lam_stv.signals.values(interval)];
lam_limit=[lam_limit; lam_lim.signals.values(interval)];
dmfdt_limit=[dmfdt_limit; dmfdt_lim.signals.values(interval)];
dmfdt_nekor=[dmfdt_nekor; dmfdt_nekorig.signals.values(interval)];
duzinaImpulsa=[duzinaImpulsa; PW_sec.signals.values(interval)]; % Trajanje ubrizgavanja [s]
v_voz=[v_voz; v_vozila.signals.values(interval)];
if imaProtokomer==1
    protokThrottleFaulty=[protokThrottleFaulty; protokThrottleLastSim];
end

% Simulacija greske "fals vazduh 6 mm". Pocetni uslovi pi2ini, T2ini i n2ini koji su dobijeni posle prvog
% postavljanja pocetnih uslova i dalje ostaju na snazi.
dOtvora=0.006; % precnik otvora u metrima
povrsinaOtvora=dOtvora^2*3.14/4;
sim(simulinkFajl, [0 simKorak:simKorak:simKraj-simKorak simKraj]);
tekucaSimulacija = tekucaSimulacija+1;
waitbar(tekucaSimulacija / ukupnoSimulacija); % Vejtbar prikazuje napredak simulacija
vreme=[vreme; tout(interval)];
ugaoLeptira=[ugaoLeptira; Angle.signals.values(interval)];
pritisakLastSim=generisiPulsacijeProc(InletPressure.signals.values(interval),odstupanjePrit);
pritisak=[pritisak; pritisakLastSim];
tempLastSim=generisiSum(InletTemp.signals.values(interval),sumTemp);
temperatura=[temperatura; tempLastSim];
revNum=[revNum; CrankshaftSpeed.signals.values(interval)];
temperaturaStvarna=[temperaturaStvarna; tempLastSim];
pritisakStvarni=[pritisakStvarni; pritisakLastSim];
protokThrottleLastSim=generisiPulsacijeProc(throttleFlow.signals.values(interval), odstupanjeProtoka);
protokThrottle=[protokThrottle;protokThrottleLastSim];
protokValve=[protokValve; portFlow.signals.values(interval)];
protokRecirk=[protokRecirk; EGRFlow.signals.values(interval)];
protokFals=[protokFals; generisiPulsacijeProc(leakFlow.signals.values(interval),odstupanjeProtoka)];
protokRecirkNF=[protokRecirkNF; EGRNFFlow.signals.values(interval)];
lambda_target=[lambda_target; lambda_zeljeno.signals.values(interval)];
lam_real=[lam_real; lam_stv.signals.values(interval)];
lam_limit=[lam_limit; lam_lim.signals.values(interval)];
dmfdt_limit=[dmfdt_limit; dmfdt_lim.signals.values(interval)];
dmfdt_nekor=[dmfdt_nekor; dmfdt_nekorig.signals.values(interval)];
duzinaImpulsa=[duzinaImpulsa; PW_sec.signals.values(interval)]; % Trajanje ubrizgavanja [s]
v_voz=[v_voz; v_vozila.signals.values(interval)];
if imaProtokomer==1
    protokThrottleFaulty=[protokThrottleFaulty; protokThrottleLastSim];
end

% Simulacija greske "EGR zagusen - 80% protok". Pocetni uslovi pi2ini, T2ini i n2ini koji su dobijeni posle
% prvog postavljanja pocetnih uslova i dalje ostaju na snazi
povrsinaOtvora=0; % ovim se iz simulacije uklanja postojanje otvora za fals vazduh
kEGR=0.8;
sim(simulinkFajl, [0 simKorak:simKorak:simKraj-simKorak simKraj]);
tekucaSimulacija = tekucaSimulacija +1;
waitbar(tekucaSimulacija / ukupnoSimulacija); % Vejtbar prikazuje napredak simulacija
vreme=[vreme; tout(interval)];
ugaoLeptira=[ugaoLeptira; Angle.signals.values(interval)];
pritisakLastSim=generisiPulsacijeProc(InletPressure.signals.values(interval),odstupanjePrit);
pritisak=[pritisak; pritisakLastSim];
tempLastSim=generisiSum(InletTemp.signals.values(interval),sumTemp);
temperatura=[temperatura; tempLastSim];
revNum=[revNum; CrankshaftSpeed.signals.values(interval)];
temperaturaStvarna=[temperaturaStvarna; tempLastSim];
pritisakStvarni=[pritisakStvarni; pritisakLastSim];
protokThrottleLastSim=generisiPulsacijeProc(throttleFlow.signals.values(interval), odstupanjeProtoka);
protokThrottle=[protokThrottle;protokThrottleLastSim];
protokValve=[protokValve; portFlow.signals.values(interval)];

```



```

protokRecirk=[protokRecirk; EGRFlow.signals.values(interval)];
protokFals=[protokFals; leakFlow.signals.values(interval)];
protokRecirkNF=[protokRecirkNF; EGRNFFlow.signals.values(interval)];
if imaProtokomer==1
    protokThrottleFaulty=[protokThrottleFaulty; protokThrottleLastSim];
end
lambda_target=[lambda_target; lambda_zeljeno.signals.values(interval)];
lam_real=[lam_real; lam_stv.signals.values(interval)];
lam_limit=[lam_limit; lam_lim.signals.values(interval)];
dmfdt_limit=[dmfdt_limit; dmfdt_lim.signals.values(interval)];
dmfdt_nekor=[dmfdt_nekor; dmfdt_nekorig.signals.values(interval)];
duzinaImpulsa=[duzinaImpulsa; PW_sec.signals.values(interval)]; % Trajanje ubrizgavanja [s]
v_voz=[v_voz; v_vozila.signals.values(interval)];

% Simulacija greske "EGR zagusen - 60 protok" . Pocetni uslovi pi2ini, T2ini i n2ini koji su dobijeni posle
% prvog postavljanja pocetnih uslova i dalje ostaju na snazi
kEGR=0.6; % Simulacija sa zagusenim EGR vodom (60% protoka)
sim(simulinkFajl, [0 simKorak:simKorak:simKraj-simKorak simKraj]);
tekucaSimulacija = tekucaSimulacija+1;
waitbar(tekucaSimulacija / ukupnoSimulacija); % Vejtbar prikazuje napredak simulacija
vreme=[vreme; tout(interval)];
ugaoLeptira=[ugaoLeptira; Angle.signals.values(interval)];
pritisakLastSim=generisiPulsacijeProc(InletPressure.signals.values(interval),odstupanjePrit);
pritisak=[pritisak; pritisakLastSim];
tempLastSim=generisiSum(InletTemp.signals.values(interval),sumTemp);
temperatura=[temperatura; tempLastSim];
revNum=generisiPulsacije([revNum; CrankshaftSpeed.signals.values(interval)],amp1RevNum);
temperaturaStvarna=[temperaturaStvarna; tempLastSim];
pritisakStvarni=[pritisakStvarni; pritisakLastSim];
protokThrottleLastSim=generisiPulsacijeProc(throttleFlow.signals.values(interval), odstupanjeProtoka);
protokThrottle=[protokThrottle;protokThrottleLastSim];
protokValve=[protokValve; portFlow.signals.values(interval)];
protokRecirk=[protokRecirk; EGRFlow.signals.values(interval)];
protokFals=[protokFals; leakFlow.signals.values(interval)];
protokRecirkNF=[protokRecirkNF; EGRNFFlow.signals.values(interval)];
if imaProtokomer==1
    protokThrottleFaulty=[protokThrottleFaulty; protokThrottleLastSim];
end
lambda_target=[lambda_target; lambda_zeljeno.signals.values(interval)];
lam_real=[lam_real; lam_stv.signals.values(interval)];
lam_limit=[lam_limit; lam_lim.signals.values(interval)];
dmfdt_limit=[dmfdt_limit; dmfdt_lim.signals.values(interval)];
dmfdt_nekor=[dmfdt_nekor; dmfdt_nekorig.signals.values(interval)];
duzinaImpulsaBezSuma=[duzinaImpulsa;PW_sec.signals.values(interval)]; % Trajanje ubrizgavanja [s]
duzinaImpulsa=generisiSum(duzinaImpulsaBezSuma,sumImpuls);
v_voz=[v_voz; v_vozila.signals.values(interval)];
if imaProtokomer==1
    protoci=[protokThrottle
        generisiPulsacijeProc([protokValve protokRecirk],odstupanjeProtoka) protokFals
        generisiPulsacijeProc(protokRecirkNF,odstupanjeProtoka) protokThrottleFaulty];
else
    protoci=[protokThrottle
        generisiPulsacijeProc([protokValve protokRecirk],odstupanjeProtoka) protokFals
        generisiPulsacijeProc(protokRecirkNF,odstupanjeProtoka)];
end

% Ovde se u matricu "protoci" dodaje na kraju kolona sa duzinom impulsa ubrizgavanja
protoci = [protoci duzinaImpulsa];
lamReal=generisiSum(lam_real,sumLambda);
lambe = [lambda_target lamReal lamReal-lam_real+lam_limit];
brzina=stepenPrenosa*ones(length(ugaoLeptira),1);
tempOkoline=Ta*ones(length(ugaoLeptira),1);

% Ovde se postavljaju vrednosti pritisak=pa, temperatura = ta, n=0, protoci=0, u slucaju da je broj obrtaja
% manji od neke minimalne vrednosti (gasenje motora)
if imaProtokomer==1 && deltaLambdaDoz<1
    temperatura = postaviLogicnuVrednost(Ta, revNum, temperatura);
    pritisak = postaviLogicnuVrednost(pa, revNum, pritisak);
    pritisakStvarni = postaviLogicnuVrednost(pa, revNum, pritisakStvarni);
% posto se matrice "protoci" i "revNum" dimenzionu ne slazu, mora se matrica "revNum" umnoziti sa repmat do
% dimenzije matrice "protoci".
koloneProtoka = size(protoci(1,:));
protoci = postaviLogicnuVrednost(0, repmat(revNum,1,koloneProtoka),protoci);
revNum = postaviLogicnuVrednost(0, revNum, revNum);
end

dataRefined=[vreme stepenToProcenat(ugaoLeptira,alphaFull,alphaNula) pritisak temperatura revNum brzina
    pritisakStvarni temperaturaStvarna tempOkoline];
if imaProtokomer==1
% Ovde se matrici dataRefined dodaje kolona sa signalom o protoku koji daje neispravan protokomer vazduha, jer
% ce ta kolona biti potrebna kasnije pri obuci neuronskih mreza za slucaj sa MAF senzorom, a dataRefined se
% memorise u excell fajl, za razliku od matrice protoci, koja se ne memorise nigde.
    dataRefined=[dataRefined protokThrottleFaulty];
end
close(h); % Zatvaranje vejtbara
if imaProtokomer==1
    if uPovSprezi==1
        fileNameBasic=['MAF-PS - Sim' num2str(simKraj) '-' num2str(simKorak)
            's - a0' num2str(alphaNula) ' - Brz.' stText ' - Temp.' num2str(Ta-273) ',Lep.'
            num2str(opterecenje) '%.xls'];
    end
end

```

```

else
    fileNameBasic=['MAF-BPS - Sim' num2str(simKraj) '-' num2str(simKorak)
        's - a0' num2str(alphaNula) ' - Brz.' stText ' - Temp.' num2str(Ta-273) ',Lep.'
        num2str(opterecenje) '%.xls'];
end
else
if uPovSprezi==1
    fileNameBasic=['SD-PS - Sim' num2str(simKraj) '-' num2str(simKorak)
        's - a0' num2str(alphaNula) ' - Brz.' stText ' - Temp.' num2str(Ta-273) ',Lep.'
        num2str(opterecenje) '%.xls'];
else
    fileNameBasic=['SD-BPS - Sim' num2str(simKraj) '-' num2str(simKorak)
        's - a0' num2str(alphaNula) ' - Brz.' stText ' - Temp.' num2str(Ta-273) ',Lep.'
        num2str(opterecenje) '%.xls'];
end
end
if svrhaSimulacije==0 % podaci za obuku
    fileNameOffered=fileNameBasic;
else % podaci za testiranje
    fileNameOffered=['Test' fileNameBasic];
end

[filename, pathname, filterindex] = uiputfile({'*.xls','Excell Files (*.xls)'}, 'Snimi rezultate simulacije kao
excell fajl',fileNameOffered);

if filename~=0
    xlswrite([pathname filename],dataRefined);
else
    errorDlg('Podaci nisu snimljeni!','Upozorenje','modal');
end

if (imaProtokomer==1)
    oznakeStanja = stanja(1); % (sa protokomerom, standardna stanja)
else
    oznakeStanja = stanja(2); % (bez protokomera, standardna stanja)
end
punoIme=[pathname filename];
iscrtajDijagrame(punoIme,dataRefined,protoci,brojStanja,oznakeStanja,1, imaProtokomer, lambde,uPovSprezi);

```

## P2.2 generisiPulsacije – izvorni kod

```

% generisiPulsacije.m

% Ova funkcija generise pulsacije na matrici podataka "dataMat", sa maks. odstupanjem +/- ampl
function pulsirajucaMatrica=generisiPulsacije(dataMat,ampl)
    dataSize = size(dataMat);
% generisanje matrice pulsacija sa pseudoslučajnim brojevima u intervalu [0, 2*ampl]
    MatPulsacija = 2*ampl*rand(dataSize);
% dodavanje matrice pulsacija na donju granicu intervala pulsiranja
    pulsirajucaMatrica = dataMat-ampl + MatPulsacija;
end

```

## P2.3 generisiPulsacijeProc – izvorni kod

```

% generisiPulsacijeProc.m

% Ova funkcija generise pulsacije na matrici podataka "dataMat", sa maksimalnim odstupanjem +/- kproc procenata
function pulsirajucaMatrica=generisiPulsacijeProc(dataMat,kproc)
    dataSize = size(dataMat);
% generisanje matrice pulsacija sa pseudoslučajnim brojevima u intervalu [0, 2*kproc/100*dataMat]
    MatPulsacija = 2*kproc/100*dataMat.*rand(dataSize);
% dodavanje matrice pulsacija na donju granicu intervala pulsiranja
    pulsirajucaMatrica = (100-kproc)/100*dataMat + MatPulsacija;
end

```

## P2.4 iscrtajDijagrame – izvorni kod

```

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
iscrtajDijagrame.m
Iscrtava opterećenje motora, temperaturu i pritisak u usisnom kolektoru, kao i broj obrtaja motora. Takodje,
iscrtava masene protoke pored leptira, pored ventila, protok fals vazduha i EGR protok
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

function iscrtajDijagrame(punoIme, dataRefined, protoci, brojStanja, oznakeStanja, zajedno, imaProtokomer,
    lambde, uPovSprezi)
    protoci=protoci*1000; % pretvaranje u grame, zbog jednostavnijeg prikaza
    bojaStv=[0.6 0 0];
    bojaNekorig=[0 0.6 0.4];
    figureViseSize=[0 0 0.85 0.9];
    figureJednaSize=[0 0 0.55 0.65];
    polozajIvelicinaOsaJedna=[0.087 0.11 0.88 0.82];
    [putanja imeFajla]=razdvojPutanjuIime(punoIme);

```

```

[brRedova,brKolona]=size(dataRefined);
intervalVertLinija = brRedova/brojStanja;
imeFajla = strrep(imeFajla, '.xls', '');

if zajedno == 1
% svi dijagrami na jednoj slici
figure('Name',[imeFajla ' -
osnovni'],'NumberTitle','off','units','normalized','outerposition',figureViseSize);
h=subaxis(2,2,4,'SH', 0.055, 'SV', 0.12, 'Padding', 0, 'ML', 0.05, 'MR', 0.02, 'MT', 0.05, 'MB', 0.07);
else
% svaki dijagram posebna slika
figure('Name',[imeFajla ' -
osnovni'],'NumberTitle','off','units','normalized','outerposition',figureJednaSize);
h=gca;
set(h,'Position',polozajIvelicinaOsaJedna);
if imaProtokomer==1
set(gcf,'CurrentAxes',h,'Name',[imeFajla ' - Ocitavanje MAF senzora'],'NumberTitle','off')
else
set(gcf,'CurrentAxes',h,'Name',[imeFajla ' - lambda'],'NumberTitle','off')
end
end

if imaProtokomer==0
plot(lambda(:,1),'LineWidth',2,'LineStyle','-','Color',bojaStv); %
hold on;
plot(lambda(:,2),'LineWidth',2,'LineStyle','-','Color','b');
hold on;
if uPovSprezi>0
plot(lambda(:,3),'LineWidth',2,'LineStyle','-','Color',bojaNekorig);
leg=legend('Zeljena','Nekorigovana','Konacna');
else
leg=legend('Zeljena','Stvarna');
end
set(leg,'Location','SouthWest');
if zajedno==1
title('(d)','FontName','Times','FontSize',14)
end
else
plot(protoci(:,1),'LineWidth',2,'LineStyle','-','Color', bojaStv);
hold on;
plot(protoci(:,6),'LineWidth',2);
leg=legend('Stvarna vrednost','Vrednost sa senzora');
set(leg,'Location','SouthWest');
if zajedno==1
title('(d)','FontName','Times','FontSize',14);
end
end
set(h,'XGrid','off','YGrid','on');
xlim([0 brRedova]);
if imaProtokomer==1
ylim([0.8*min(protoci(:,6)) 1.35*max(protoci(:,6))]);
vrhDijagrama=get(h,'YLim');
yTeksta=0.9*vrhDijagrama(1,2);
else
ylim([0 1.6*max(max(lambda))]);
vrhDijagrama=get(h,'YLim');
yTeksta=0.92*vrhDijagrama(1,2);
end
set(h,'XTick',0:intervalVertLinija:brRedova);
oznaciStanja(h,intervalVertLinija,brRedova,yTeksta,oznakeStanja,10,2,false);
xlabel('Broj uzorka','FontName','Times','FontSize',12);
if imaProtokomer==0
ylabel(['\lambda' ' \rm[-]'],'FontName','Times','FontSize',12)
else
ylabel(['$\dot{m}_{at}$' '
\rm[g/s]'],'FontName','Times','FontSize',13,'FontWeight','bold','interpreter','latex');
end

% ako je svaki dijagram nacrtan posebno (bez subplota), svaki od njih se snima u fajl
if zajedno == 0
if imaProtokomer==0
hgsave(gcf,[putanja imeFajla ' - lambda.fig'])
else
hgsave(gcf,[putanja imeFajla ' - MAF signal.fig'])
end
end

if zajedno == 1
h=subaxis(2,2,3,'SH', 0.045, 'SV', 0.12, 'Padding', 0, 'ML', 0.05, 'MR', 0.01, 'MT', 0.05, 'MB', 0.07);
else
figure('Name',[imeFajla ' - Temperatura'],'NumberTitle','off','units','normalized','outerposition',
figureJednaSize);
h=gca;
set(h,'Position',polozajIvelicinaOsaJedna);
hgsave(gcf,[putanja imeFajla ' - Temperatura.fig']);
end

if imaProtokomer==0
plot(dataRefined(:,8)-273,'LineWidth',2,'LineStyle','--','Color',bojaStv);
hold on;

```

```

end
plot(dataRefined(:,4)-273,'LineWidth',2);
if imaProtokomer==0
    leg=legend('Stvarna vrednost','Vrednost sa senzora');
    set(leg,'Location','South');
end

if zajedno==1
    title('c','FontName','Times','FontSize',14);
end
set(h,'XGrid','off','YGrid','on');
xlim([0 brRedova]);
ylim([(min(dataRefined(:,4))-273)-10 (max(dataRefined(:,4))-273)+15]);
vrhDijagrama=get(h,'YLim');
set(h,'XTick',0:intervalVertLinija:brRedova);
oznaciStanja(h,intervalVertLinija,brRedova,0.88*vrhDijagrama(1,2),oznakeStanja,10,2,false);
xlabel('Broj uzorka','FontName','Times','FontSize',12);
ylabel(['t_i ' \rm[ ^{o}C]],'FontName','Times','FontSize',12)
% ako je svaki dijagram nacrtan posebno (bez subplota), svaki od njih se snima u fajl
if zajedno == 0
    hgsave(gcf,[putanja imeFajla ' - Temperatura.fig']);
end
if zajedno == 1
    h=subaxis(2,2,1,'SH', 0.055, 'SV', 0.12, 'Padding', 0, 'ML', 0.05, 'MR', 0.02, 'MT', 0.05, 'MB', 0.07);
else
    figure('Name',[imeFajla ' - Pritisak'],'NumberTitle','off','units','normalized','outerposition',figureJednaSize);
    h=gca;
    set(h,'Position',polozajIvelicinaOsaJedna);
    hgsave(gcf,[putanja imeFajla ' - Pritisak.fig']);
end

% Koeficijent za pretvaranje bara u kilopaskale
barTokPaKoeff=100;
plot(dataRefined(:,7)*barTokPaKoeff,'LineWidth',2,'LineStyle','-','Color',bojaStv); %
hold on;
plot(dataRefined(:,3)*barTokPaKoeff,'LineWidth',2);
leg=legend('Stvarna vrednost','Vrednost sa senzora');
set(leg,'Location','SouthEast');
if zajedno==1
    title('a','FontName','Times','FontSize',14);
end
set(h,'XGrid','off','YGrid','on');
xlim([0 brRedova]);
ylim([0.0*min(dataRefined(:,3))*barTokPaKoeff 1.5*barTokPaKoeff*max(dataRefined(:,3))]);
vrhDijagrama=get(h,'YLim');
set(h,'XTick',0:intervalVertLinija:brRedova);
oznaciStanja(h,intervalVertLinija,brRedova,0.92*vrhDijagrama(1,2),oznakeStanja,10,2,false);
xlabel('Broj uzorka','FontName','Times','FontSize',12);
ylabel(['p_i ' \rm[kPa]],'FontName','Times','FontSize',12)

% ako je svaki dijagram nacrtan posebno (bez subplota), svaki od njih se snima u fajl
if zajedno == 0
    hgsave(gcf,[putanja imeFajla ' - Pritisak.fig']);
end
if zajedno == 1
    h=subaxis(2,2,2,'SH', 0.045, 'SV', 0.12, 'Padding', 0, 'ML', 0.05, 'MR', 0.01, 'MT', 0.05, 'MB',
    0.07);
else
    figure('Name',[imeFajla ' - Broj obrtaja'],'NumberTitle','off','units','normalized','outerposition',figureJednaSize);
    h=gca;
    set(h,'Position',polozajIvelicinaOsaJedna);
    hgsave(gcf,[putanja imeFajla ' - Broj obrtaja.fig']);
end
plot(dataRefined(:,5)*1000,'LineWidth',2);
if zajedno==1
    title('b','FontName','Times','FontSize',14);
end
set(h,'XGrid','off','YGrid','on');
xlim([0 brRedova]);
ylim([0.95*min(dataRefined(:,5)*1000) 1.35*max(dataRefined(:,5)*1000)]);
vrhDijagrama=get(h,'YLim');
set(h,'XTick',0:intervalVertLinija:brRedova);
oznaciStanja(h,intervalVertLinija,brRedova,0.93*vrhDijagrama(1,2),oznakeStanja,10,2,false);
xlabel('Broj uzorka','FontName','Times','FontSize',12);
ylabel(['n ' \rm[1/min]],'FontName','Times','FontSize',12)

% ako je svaki dijagram nacrtan posebno (bez subplota), svaki od njih se snima u fajl
if zajedno == 0
    hgsave(gcf,[putanja imeFajla ' - Broj obrtaja.fig']);
end
if zajedno == 1
    hgsave(gcf,[putanja imeFajla ' - osnovni.fig']);
end
figure('Name',[imeFajla ' - protoci'],'NumberTitle','off','units','normalized','outerposition',figureViseSize);
h=subaxis(2,2,1,'SH', 0.055, 'SV', 0.12, 'Padding', 0, 'ML', 0.05, 'MR', 0.02, 'MT', 0.05, 'MB', 0.07);
plot(protoci(:,1),'LineWidth',2);
title('Maseni protok vazduha pored leptira','FontName','Times','FontSize',14);

```

```

set(h,'XGrid','off','YGrid','on');
xlim([0 brRedova]);
ylim([0.96*min(protoci(:,1)) 1.4*max(protoci(:,1))]);
vrhDijagrama=get(h,'YLim');
set(h,'XTick',0:intervalVertLinija:brRedova);
oznaciStanja(h,intervalVertLinija,brRedova,0.93*vrhDijagrama(1,2),oznakeStanja,10,2,false);
xlabel('Broj uzorka','FontName','Times','FontSize',12);
ylabel(['$\dot{m}_{at}$' '\rm[g/s]'],'FontName','Times','FontSize',13,'FontWeight','bold','interpreter','latex');

h=subaxis(2,2,2,'SH', 0.045, 'SV', 0.12, 'Padding', 0, 'ML', 0.05, 'MR', 0.01, 'MT', 0.05, 'MB', 0.07);
plot(protoci(:,2),'LineWidth',2);
title('Ukupni maseni protok kroz usisni kolektor','FontName','Times','FontSize',14);
set(h,'XGrid','off','YGrid','on');
xlim([0 brRedova]);
ylim([0.97*min(protoci(:,2)) 1.4*max(protoci(:,2))]);
vrhDijagrama=get(h,'YLim');
set(h,'XTick',0:intervalVertLinija:brRedova);
oznaciStanja(h,intervalVertLinija,brRedova,0.93*vrhDijagrama(1,2),oznakeStanja,10,2,false);
xlabel('Broj uzorka','FontName','Times','FontSize',12);
ylabel(['$\dot{m}_{ap}$' '\rm[g/s]'],'FontName','Times','FontSize',13,'FontWeight','bold','interpreter','latex');

h=subaxis(2,2,3,'SH', 0.055, 'SV', 0.12, 'Padding', 0, 'ML', 0.05, 'MR', 0.02, 'MT', 0.05, 'MB', 0.07);
plot(protoci(:,3),'LineWidth',2);
title('Maseni EGR protok','FontName','Times','FontSize',14);
set(h,'XGrid','off','YGrid','on');
xlim([0 brRedova]);
ylim([0.9*min(protoci(:,3)) 1.35*max(protoci(:,3))]);
vrhDijagrama=get(h,'YLim');
set(h,'XTick',0:intervalVertLinija:brRedova);
oznaciStanja(h,intervalVertLinija,brRedova,0.928*vrhDijagrama(1,2),oznakeStanja,10,2,false);
xlabel('Broj uzorka','FontName','Times','FontSize',12);
ylabel(['$\dot{m}_{EGR}$' '\rm[g/s]'],'FontName','Times','FontSize',13,'FontWeight','bold','interpreter','latex');

h=subaxis(2,2,4,'SH', 0.045, 'SV', 0.12, 'Padding', 0, 'ML', 0.05, 'MR', 0.01, 'MT', 0.05, 'MB', 0.07);
plot(protoci(:,4),'LineWidth',2);
title('Maseni protok fals vazduha','FontName','Times','FontSize',14);
set(h,'XGrid','off','YGrid','on');
xlim([0 brRedova]);
ylim([0.85*min(protoci(:,4)) 1.6*max(protoci(:,4))]);
vrhDijagrama=get(h,'YLim');
set(h,'XTick',0:intervalVertLinija:brRedova);
oznaciStanja(h,intervalVertLinija,brRedova,0.915*vrhDijagrama(1,2),oznakeStanja,10,2,false);
xlabel('Broj uzorka','FontName','Times','FontSize',12);
ylabel(['$\dot{m}_{fv}$' '\rm[g/s]'],'FontName','Times','FontSize',13,'interpreter','latex');
hgsave(gcf,[putanja imeFajla ' - protoci.fig']);

if imaProtokomer==1
    figure('Name',[imeFajla ' - lambda'],'NumberTitle','off','units','normalized','outerposition',figureJednaSize);
    h=gca;
    set(h,'Position',polozajIvelicinaOsaJedna);
    plot(lambda(:,1),'LineWidth',2,'LineStyle','-','Color',bojaStv); %
    hold on;
    plot(lambda(:,2),'LineWidth',2,'LineStyle','-','Color','b');
    hold on;
    if uPovSprezi>0
        plot(lambda(:,3),'LineWidth',2,'LineStyle','-','Color',bojaNekorig);
        leg=legend('Zeljena','Nekorigovana','Konacna');
    else
        leg=legend('Zeljena','Stvarna');
    end
    set(leg,'Location','SouthWest');
    set(h,'XGrid','off','YGrid','on');
    xlim([0 brRedova]);
    ylim([0 1.6*max(max(lambda))]);
    vrhDijagrama=get(h,'YLim');
    set(h,'XTick',0:intervalVertLinija:brRedova);
    oznaciStanja(h,intervalVertLinija,brRedova,0.92*vrhDijagrama(1,2),oznakeStanja,10,2,false);
    xlabel('Broj uzorka','FontName','Times','FontSize',12);
    ylabel(['\lambda' '\rm[-]'],'FontName','Times','FontSize',12);
    hgsave(gcf,[putanja imeFajla ' - lambda.fig']);
end
end

```

## P2.5 oznaciStanja – izvorni kod

```
% oznaciStanja.m
```

```
% Ova funkcija iscratava vertikalne linije, koje omogućavaju da se jasnije vide pojedina stanja neispravnosti
na vec nacrtanim dijagramima velicina koje se prate (npr. opterećenje motora, pritisak i temperatura u
usisnom kolektoru, broj obrtaja). Osim toga, upisuje i oznake pojedinih stanja na odgovarajucim pozicijama.
Parametri koji se prosledjuju ovoj funkciji su:
% currentAxes - aktuelni koordinatni sistem,
% intervalVertLinija - razmak izmedju susednih vertikalnih linija,
% matricaVrednosti - matrica sa podacima o velicini na cijem dijagramu se oznacavaju stanja,
```

```

% brRedova - ukupan broj redova matriceVrednosti tj. ukupan broj semplova u simulacijama u jednoj radnoj
% tacki pomocu fajla MVEMSimStepeniPrenosa.m
% merodavnaKolona - indeks kolone u matriciVrednosti u kojoj se nalazi merodavna velicina
% koeficijentPolozaja - realan broj, pomocu kojeg se lociraju oznake stanja iznad glavne linije dijagrama
% stanja - niz stringova sa oznakama pojedinih stanja
% velicinaSlova - velicina teksta koji na dijagramu oznacava stanja
% tekstIndent - integer koji oznacava pocetak ispisivanja oznake stanja na dijagramu

function oznaciStanja(currentAxes,intervalVertLinija,brRedova,yPozicijaTeksta,stanja,
                    velicinaSlova,tekstIndent,NN)

if NN==1
    OKBoja=[0 0 0]; % crna
    FaultIBoja=[0.9 0 0]; %crvena osnova
    FaultIIBoja=[0 0.7 0]; %zelena osnova
    FaultIIIBoja=[0 0 0.9]; % plava osnova
    FaultIVBoja=[0.9 0 0.9]; % ljubicasta osnova
end
hax=currentAxes;
visina=get(hax,'YLim');
brojac=1;
for i=0:intervalVertLinija:brRedova
    line([i i],visina,'Color',[0 0 0],'LineStyle','-');
    if i<brRedova
        duzinaOznake = length(stanja{brojac});
        switch duzinaOznake
            case 2
                xPozicijaTeksta = tekstIndent+3;
            case 3
                xPozicijaTeksta = tekstIndent+2;
            otherwise
                xPozicijaTeksta = tekstIndent;
        end
    end

    if NN
        switch brojac
            case 1
                bojaTeksta=OKBoja;
            case {2,3,4,5}
                bojaTeksta=FaultIBoja;
            case {6,7,8,9}
                bojaTeksta=FaultIIBoja;
            case {10,11,12}
                bojaTeksta=FaultIIIBoja;
            otherwise
                bojaTeksta=FaultIVBoja;
        end
    else
        bojaTeksta=[1 0 0];
        xPozicijaTeksta=xPozicijaTeksta-1;
    end
    text(i+xPozicijaTeksta,yPozicijaTeksta,stanja{brojac},'FontSize',velicinaSlova,'Color',
        bojaTeksta,'FontWeight','normal','FontName','Cambria') % vertikalni polozaj teksta fiksna
end
    brojac=brojac+1;
end
end
end

```

## P3 Izvorni kodovi funkcija napisanih u okviru razvoja simulacionog dijagnostičkog modela

### P3.1 MatricaZeljenihIzlaza – izvorni kod

```
% MatricaZeljenihIzlaza.m

% Ovo je funkcija za generisanje matrice zeljenih izlaza neuronske mreze, formata mxn,
% pri cemu je n ukupan broj semplova, a m je broj stanja usisnog sistema (brKol),
% ukljucujuci i stanje bez neispravnosti. Broj semplova u jednoj simulaciji je
% brSemplova i on pokazuje koliko ce uzastopnih jedinica biti u svakoj koloni izlazne matrice.

function target=MatricaZeljenihIzlaza(brKol,brSemplova)
    modulMat=ones(brSemplova,1);
    for i=1:brKol-1
        modulMat=blkdiag(modulMat,ones(brSemplova,1));
    end
    target=modulMat';
end
```

### P3.2 generisiStrukturuZaObuku – izvorni kod

```
% generisiStrukturuZaObuku.m

% Ova funkcija generise jednu Matlab strukturu, koja sadrzi matricu za obuku dobijenu na osnovu rezultata
% simulacija za razlicita opterecenja motora i razlicite temperature okoline. Podaci za generisanje ove matrice
% preuzimaju se iz odgovarajucih excell fajlova. Osim toga, ova struktura sadrzi i target matricu koja odgovara
% generisanoj matrici za obuku, pri cemu je pomocu ulaznog parametra "brojStanja" uzet u obzir broj razlicitih
% stanja koje neuronska mreza treba kasnije da detektuje. Broj semplova u svakom xls fajlu, koji odgovara svim
% simulacijama u jednoj radnoj tacki, predstavljen je promenljivom "brojSemplova". Inace, naziv "TTStruktura"
% potice od T(raining)T(Struktura i oznacava da je u pitanju Matlab struktura (tip podatka Structure),
% koja sadrzi matricu za obuku (Training) i ciljnu matricu (Target)

function TTstruktura=generisiStrukturuZaObuku(brojStanja,normalizujTrainMat,stPrenosa)
    filenames=uipickfiles('FilterSpec','*.xls');
    if ~isempty(filenames)
        if iscell(filenames) %Ako je filename Cell of arrays, izvodi se procedura za multiselect slucaj
            duzina=length(filenames);
            Temp=cell(3,1);
            Temp{3,1}=[];
            punoImePrvogFajla=filenames{1};
            [putanjaPrvog imePrvog]=razdvojPutanjuIime(punoImePrvogFajla);
            pocetakImenaFajla=imePrvog(1:strfind(imePrvog,'Sim')-1);
            imeFajla=[pocetakImenaFajla 'Brzina' num2str(stPrenosa) '. '];
            for i=1:duzina
                putanjaIime=filenames{i};
                [putanja tempImeFajla]=razdvojPutanjuIime(putanjaIime);
                fileRead=xlsread([putanja tempImeFajla]);
                imeBezEkstenzije=strrep(tempImeFajla,'.xls','');
                odKaraktera=strfind(imeBezEkstenzije,'Temp. ');
                doKaraktera=length(imeBezEkstenzije);
                Temp{i,1}=fileRead;
                tempOptDeoImena=imeBezEkstenzije(odKaraktera:doKaraktera);
                if (i==1)
                    skraceniTempOptDeoImena=strrep(tempOptDeoImena,'Temp.','T');
                    skraceniTempOptDeoImena=strrep(skraceniTempOptDeoImena,'Lep.','L');
                else
                    skraceniTempOptDeoImena=strrep(tempOptDeoImena,'Temp.','');
                    skraceniTempOptDeoImena=strrep(skraceniTempOptDeoImena,'Lep.','');
                end
                skraceniTempOptDeoImena=strrep(skraceniTempOptDeoImena,'%','');
                imeFajla=[imeFajla ' -' skraceniTempOptDeoImena];
            end
            TTstruktura.trainMat=cell2mat(Temp);
            % Brisu se kolone matrice koja sadrzi podatke o tekucem vremenu simulacija, stepenu prenosu, stvarnom pritisku
            % i stvarnoj temperaturi u usisu, jer ti podaci nisu potrebni za obucavanje neuronske mreze
            TTstruktura.trainMat(:,6:8)=[];
            TTstruktura.trainMat(:,1)=[];
            % Prebacivanje u format potreban za neuronsku mrezu, gde je broj vrsta jednak broju ulaza u neuronsku mrezu, a
            % broj kolona jednak broju semplova
            TTstruktura.trainMat=TTstruktura.trainMat';
            % U novoj verziji broj radnih tacaka ce zameniti broj polozaaja leptira, jer se isti polozej leptira moze
            % ponoviti vise puta, ako se selektuje vise fajlova (isti polozej leptira, a razlicite temperature vazduha)
            TTstruktura.brojRadnihTacaka=duzina;
            % Imenu fajla dodaje se verzija Matlaba u kojoj je fajl generisan (ovo je interno potrebno zbog provere
            % kompatibilnosti)
            TTstruktura.imeTrainingFajla=[imeFajla ' (' version('-release') ') '];
        end
    end
end
```

```

    brojSemplova=length(TTstruktura.trainMat(1,:))/brojStanja/TTstruktura.brojRadnihTacaka; % brojSemplova
    je broj semplova u matrici za obuku pri simulaciji jednog stanja

% Sada se ciljna matrica za jedan položaj leptira replicira onoliko puta koliko ima različitih radnih tacaka
    tj. koliko je fajlova izabrano u multiselekt opciji
    TTstruktura.targetMat= repmat(MatricaZeljnihIzlaza(brojStanja,brojSemplova),
        1,TTstruktura.brojRadnihTacaka);
    nazivFajlaStrukture=['TT' num2str(brojStanja) imeFajla '.mat'];
    else % Ako filename nije Cell of arrays, izvodi se procedura za singleselect slucaj
    TTstruktura.trainMat=xlswread([pathname filename]);
% Brise se prva kolona matrice koja sadrzi podatke o tekucem vremenu simulacija,
% jer ti podaci nisu potrebni za obucavanje neuronske mreze
    TTstruktura.trainMat(:,1)=[];
% Prebacivanje u format potreban za neuronsku mrežu, gde je broj vrsta jednak broju ulaza u neuronsku mrežu, a
    broj kolona jednak broju semplova
    TTstruktura.trainMat=TTstruktura.trainMat';
    if normalizujTrainMat
    TTstruktura.trainMat=normr(TTstruktura.trainMat); % normalizuju se redovi matrice za obuku
    end
    TTstruktura.brojRadnihTacaka=1;
    TTstruktura.imeTrainingFajla=[filename (' version('-release') ')'];
    brojSemplova=length(TTstruktura.trainMat(1,:))/brojStanja/TTstruktura.brojRadnihTacaka; % brojSemplova
    je broj semplova u matrici za obuku pri simulaciji jednog stanja
    TTstruktura.targetMat= repmat(MatricaZeljnihIzlaza(brojStanja,brojSemplova),
        1,TTstruktura.brojRadnihTacaka);
    nazivFajlaStrukture=['TT' num2str(brojStanja) strrep(filename, '.xls','') (' version('-release') ') '
        '.mat'];
end
if normalizujTrainMat
    TTstruktura.trainMat=normr(TTstruktura.trainMat);
    nazivFajlaStrukture=['Norm' nazivFajlaStrukture];
end
trainMat=TTstruktura.trainMat;
brojRadnihTacaka=TTstruktura.brojRadnihTacaka;
TTstruktura.imeTrainingFajla=nazivFajlaStrukture;
imeTrainingFajla=TTstruktura.imeTrainingFajla;
targetMat=TTstruktura.targetMat;
[fajlStrukture, putanjaStrukture, filterindex] = uiputfile( ...
    {'*.mat','Matlab fajlovi (*.mat)'},...
    'Snimite podatke potrebne za generisanje neuronske mreze',nazivFajlaStrukture);
if fajlStrukture~=0
    save (fajlStrukture,'trainMat','brojRadnihTacaka','targetMat','imeTrainingFajla');
else
    error('Podaci o ulaznoj strukturi nisu snimljeni u fajl!','Upozorenje!','modal');
end
else
    error('Niste izabrali fajlove. Operacija se ne moze izvršiti.','Greska!','modal');
end
return;
end
end

```

### P3.3 kreirajNetStrukturu – izvorni kod

```

% kreirajNetStrukturu.m

% Ovde se ucitavaju podaci za obucavanje i odgovarajuca target matrica. Onda se generise MLP neuronska mreza,
    gde je "nBroj" matrica cija svaka kolona predstavlja po jedan skriveni sloj, a vrednosti u kolonama oznacavaju
    broj neurona u odgovarajućem skrivenom sloju. Preko parametra "greska" moze se zadati velicina greske koja se
    zeli postici prilikom obuke neuronske mreze.

function NNSstruktura = kreirajNetStrukturu(brojStanja,greska,nBroj,iterationsNum, showInterval,
    pragDetekcije,imaMAF)
% Odabir fajla koji sadrzi strukturu sa matricom za obuku i odgovarajućom matricom zeljenih izlaza
    TTstruktura=ucitajTTstrukturu;
    TrainingMat=TTstruktura.trainMat;
    TargetMat=TTstruktura.targetMat;
    TrainMatFileName=TTstruktura.imeTrainingFajla;
% Generisanje neuronske mreze
    mreza = newpr(TrainingMat,TargetMat,nBroj);
    mreza.trainParam.epochs=iterationsNum;
    mreza.trainParam.goal=greska;
    mreza.trainParam.show=showInterval;
    tStart=tic; % trenutak pocetka obuke
    mreza=init(mreza);
    [mreza, tr]=train(mreza,TrainingMat,TargetMat);
    tStop=toc(tStart); % trenutak zavrsetka obuke
    brojNNSlojeva=length(nBroj);
    NNSstruktura.podaciOMrezi.brojSlojeva=brojNNSlojeva;
    oznakaBrojaNN=' ';
    for i=1:brojNNSlojeva
        NNSstruktura.podaciOMrezi.neuronaPoSloju(i)=nBroj(i);
        oznakaBrojaNN=[oznakaBrojaNN num2str(nBroj(i))];
        if i<brojNNSlojeva
            oznakaBrojaNN=[oznakaBrojaNN ','];
        end
    end
end

TestStruktura=generisiTestStrukturu;
netIzlaz=sim(mreza,TestStruktura.matrica);

```



```

NNStruktura.podaciOMrezi.mreza=mreza;
NNStruktura.podaciOMrezi.targetMatrica=TargetMat;
NNStruktura.podaciOMrezi.obuka=tr;
NNStruktura.podaciOMrezi.brojStanja=brojStanja;
NNStruktura.podaciOMrezi.trajanje=tStop;
NNStruktura.podaciOMrezi.matricaZaObuku=TrainingMat;
NNStruktura.podaciOMrezi.imeTreningFajla=TrainMatFileName;
NNStruktura.podaciORezultatima.testMatrica=TestStruktura.matrica;
NNStruktura.podaciORezultatima.imeTestingFajla=TestStruktura.imeFajla;
NNStruktura.podaciORezultatima.putanjaTestingFajla=TestStruktura.putanja;
NNStruktura.podaciORezultatima.izlaznaMatrica=netIzlaz;
NNStruktura.podaciORezultatima.maxGreska=greska;
NNStruktura.podaciORezultatima.greska=min(NNStruktura.podaciOMrezi.obuka.perf);
minGreska=NNStruktura.podaciORezultatima.greska;

namePrefiks=['MLP' oznakaBrojaNN ' G' num2str(minGreska)];
[fajlMreze, putanjaMreze, filterindex] = ...
    uiputfile({'*.mat','Matlab fajlovi (*.mat)'},'Snimi podatke o neuronskoj mrezi kao mat-fajl',
        [namePrefiks ' Obuka ']' strrep(TrainMatFileName,'.mat','') ' (' version('-release') ') .mat']);
if fajlMreze~=0
    NNStruktura.podaciOMrezi.fajlMreze=fajlMreze;
    save (fajlMreze,'NNStruktura');
else
    error('Podaci o neuronskoj mrezi nisu snimljeni!','Upozorenje!','modal');
end

prviRedNaslava=['Mreza: ' namePrefiks ' , Ciljna greska: ' num2str(greska)];
naslovDijagrama={prviRedNaslava; ['Obuka iz fajla: ' TrainMatFileName]; ['Testiranje iz fajla: '
    TestStruktura.imeFajla];
xlabela='Broj uzorka';
ylabela='Izlaz neuronske mreze';

% Vrsta simulacije: 1 (sa protokomera, standardna stanja); 2 (bez protokomera, standardna stanja), 3 (sa
protokomera, nepoznata stanja), 4 (bez protokomera, nepoznata stanja)
if (imaMAF==1)
    oznakeStanja = stanjaNN(1); % (sa protokomera, standardna stanja)
else
    oznakeStanja = stanjaNN(2); % (bez protokomera, standardna stanja)
end
iscrtajNNizlaz (netIzlaz,pragDetekcije,naslovDijagrama,xLabela,yLabela,brojStanja,oznakeStanja,fajlMreze);
end

```

### P3.4 generisiTestStrukturu – izvorni kod

```

% generisiTestStrukturu.m
% Pomocu ovog skripta ucitavaju se podaci za testiranje neuronske mreze iz xls fajla.
function testStrukt = generisiTestStrukturu
% Ucitavanje matrice za testiranje
[imeTestingFajla, putanjaTestingFajla] = uigetfile( ...
    {'*.xls','Excell Files (*.xls)'},'Izaberite fajl sa podacima za testiranje');
if imeTestingFajla~=0
    matricaZaTestiranje=xlsread([putanjaTestingFajla imeTestingFajla]);
% Posle ucitavanja, brisi se kolone matrice koje sadrze podatke o tekucem vremenu simulacija, stepenu prenosa,
stvarnom pritisku i stvarnoj temperaturi u usisu, jer ti podaci nisu potrebni za obucavanje neuronske mreze.
    matricaZaTestiranje(:,6:8)=[];
    matricaZaTestiranje(:,1)=[];
else
    matricaZaTestiranje=0;
    error('Ne moze se izvrsiti testiranje neuronske mreze, ako ne ucitate podatke za
        testiranje','Greska!','modal');
return;
end
% Transponovanje matrice za testiranje u format koji je neophodan za neuronsku mrezu
matrica=matricaZaTestiranje';
testStrukt.matrica=matrica;
testStrukt.imeFajla=imeTestingFajla;
testStrukt.putanja=putanjaTestingFajla;
end

```

### P3.5 testirajNetStrukturu – izvorni kod

```

% testirajNetStrukturu.m
% Ovde se testira struktura koja u svom sastavu sadrzi neuronsku mrezu. Parametar "pragDetekcije" je granica
koja omogućava sistemu da prepozna neki događaj. Ako je izlaz iz mreze veci od "pragDetekcije", onda se
desio taj događaj. Bice uzeta vrednost pragDetekcije=0.5. Parametar nepozGres moze imati vrednosti: 0
(testiranje poznatih neispravnosti), 1 (testiranje nepoznatih neispravnosti, varijanta 1) ili 2 (testiranje
nepoznatih neispravnosti, varijanta 2) ili 3 (testiranje nepoznatih neispravnosti, varijanta 3)

function testiranaNetStruktura=testirajNetStrukturu (pragDetekcije, imaMAF, nepozGres,
    viseGresaka, brojUzastopnihSemplova)

[imeStrukFajla, putanjaStrukFajla] = uigetfile( ...
    {'*.mat','Matlab Files (*.mat)'},'Izaberite fajl koji sadrzi neuronsku mrezu');
if imeStrukFajla~=0
% Ucitavanje postojece mreze
    testiranaNetStruktura=importdata([putanjaStrukFajla imeStrukFajla]);
    mreza=testiranaNetStruktura.podaciOMrezi.mreza;

```

```

fajlMreze=testiranaNetStruktura.podaciOMrezi.fajlMreze;
brojStanja=testiranaNetStruktura.podaciOMrezi.brojStanja;
% Ucitavanje podataka koji ce poslužiti za testiranje
TestStruktura=generisiTestStrukturu;
% Primena ucitane neuronske mreze na podatke za testiranje
NetIzlaz=sim(mreza, TestStruktura.matrica);
% Upisivanje novih podataka o rezultatima u strukturu koja sadrzi neuronsku mrezu
testiranaNetStruktura.podaciORezultatima.testMatrica=TestStruktura.matrica;
testiranaNetStruktura.podaciORezultatima.imeTestingFajla=TestStruktura.imeFajla;
testiranaNetStruktura.podaciORezultatima.putanjaTestingFajla=TestStruktura.putanja;
testiranaNetStruktura.podaciORezultatima.izlaznaMatrica=NetIzlaz;
preradjenNetIzlaz = preradiRedukujNNizlaz (NetIzlaz,brojUzastopnihSemplova, viseGresaka,brojStanja);
testiranaNetStruktura.podaciORezultatima.izlaznaFiltriranaMatrica=preradjenNetIzlaz;

skraceniNazivTestFajla=TestStruktura.imeFajla;
odKaraktera=strfind(skraceniNazivTestFajla,'Temp.');
```

```

doKaraktera=length(skraceniNazivTestFajla);
skraceniNaziv=skraceniNazivTestFajla(odKaraktera:doKaraktera);
skraceniNaziv=strrep(skraceniNaziv,'Temp.','T');
skraceniNaziv=strrep(skraceniNaziv,'Opter.','O');
skraceniNaziv=strrep(skraceniNaziv,'%.xls','');
```

```

if viseGresaka==true
    sufiks='';
else
    sufiks='-Jedna';
end

if (nepozGres>0)
    nazivFajlaStrukture=[strrep(imeStrukFajla,'.mat','') ',' skraceniNaziv ' ' sufiks ' NG'
                        num2str(nepozGres) '''.mat'];
else
    nazivFajlaStrukture=[strrep(imeStrukFajla,'.mat','') ',' skraceniNaziv ' ' sufiks '''.mat'];
end
fajlStrukture = uiputfile( ...
    {'*.mat','Matlab fajlovi (*.mat)'},...
    'Snimi nove podatke o neuronskoj mrezi kao mat-fajl',nazivFajlaStrukture);
if fajlStrukture~=0
    save (fajlStrukture,'testiranaNetStruktura');
else
    errordlg('Podaci o neuronskoj mrezi nisu snimljeni!','Upozorenje!','modal');
end

brojNNSlojeva=testiranaNetStruktura.podaciOMrezi.brojslojeva;
oznakaBrojaNN=' ';
for i=1:brojNNSlojeva
    nBroj(i)=testiranaNetStruktura.podaciOMrezi.neuronaPoSloju(i);
    oznakaBrojaNN=[oznakaBrojaNN num2str(nBroj(i))];
    if i<brojNNSlojeva
        oznakaBrojaNN=[oznakaBrojaNN ' '];
    end
end
namePrefiks=['MLP' oznakaBrojaNN];
maxGreska=testiranaNetStruktura.podaciORezultatima.maxGreska;
greska=testiranaNetStruktura.podaciORezultatima.greska;
TrainMatFileName=testiranaNetStruktura.podaciOMrezi.imeTreningFajla;
prviRedNaslava=['Mreza: ' namePrefiks ' , Ciljna greska: ' num2str(maxGreska) ' , Stvarna greska: '
                num2str(greska)];
naslovDijagrama={prviRedNaslava; ['Obuka iz fajla: ' TrainMatFileName]; ['Testiranje iz fajla: '
                                TestStruktura.imeFajla]};
xLabela='Broj uzorka';
yLabela='Izlaz neuronske mreze';

if (imaMAF==1)
    if viseGresaka
        switch nepozGres
            case 0
                oznakeStanja = stanjaNN(1); % (sa protokomerom, standardna stanja)
            case 1
                oznakeStanja = stanjaNN(3); % (sa protokomerom, nepoznata stanja I)
            case 2
                oznakeStanja = stanjaNN(5); % (sa protokomerom, nepoznata stanja II)
            case 3
                oznakeStanja = stanjaNN(7); % (sa protokomerom, nepoznata stanja III)
            case 4
                oznakeStanja = stanjaNN(9); % (sa protokomerom, nepoznata stanja IV)
        end
    else
        oznakeStanja = stanjaNN(1); % (sa protokomerom, standardna stanja)
    end
else
    if viseGresaka
        switch nepozGres
            case 0
                oznakeStanja = stanjaNN(2); % (bez protokomera, standardna stanja)
            case 1
                oznakeStanja = stanjaNN(4); % (bez protokomera, nepoznata stanja I)
            case 2
                oznakeStanja = stanjaNN(6); % (bez protokomera, nepoznata stanja II)
            case 3
                oznakeStanja = stanjaNN(8); % (bez protokomera, nepoznata stanja III)
        end
    end
end
end

```

```

        oznakeStanja = stanjaNN(8); % (bez protokomera, nepoznata stanja III)
    case 4
        oznakeStanja = stanjaNN(10); % (bez protokomera, nepoznata stanja IV)
    end
    else
        oznakeStanja = stanjaNN(2); % (bez protokomera, standardna stanja)
    end
end
if (viseGresaka==true)
    iscrtajNNizlaz(NetIzlaz, pragDetekcije, naslovDijagrama, xLabela, yLabela,
        brojStanja, oznakeStanja, fajlStrukture);
    iscrtajNNizlaz(preradjenNetIzlaz, pragDetekcije, naslovDijagrama, xLabela, yLabela,
        brojStanja, oznakeStanja, [fajlStrukture '-f']);
else
    iscrtajRedoveBojeSingle(NetIzlaz, pragDetekcije, naslovDijagrama, xLabela, yLabela,
        oznakeStanja, fajlStrukture);
    iscrtajRedoveBojeSingle(preradjenNetIzlaz, pragDetekcije, naslovDijagrama, xLabela, yLabela,
        oznakeStanja, [fajlStrukture '-f']);
end
end
end
end

```

## P3.6 preradiRedukujNNizlaz – izvorni kod

```

% preradiRedukujNNizlaz.m

% Ova funkcija omogućava da se u neku ruku isfiltrira izlaz iz neuronske mreze, jer postoji veliki broj
% pojedinačnih pikova koji mogu da dovedu do laznog alarma. Parametri funkcije su matrica "NNizlaz", koja treba
% da se preradi, zatim broj uzastopnih izlaznih vrednosti koje treba uproseciti (integer "n"), i "viseGresaka" je
% logicka promenljiva kojom se oznacava da li matrica "NNizlaz" sadrzi rezultate dijagnostikovanja neke
% pojedinačne greske ili svih simuliranih gresaka.
function preradjenIzlaz = preradiRedukujNNizlaz (NNizlaz, n, viseGresaka, brojStanja)
    [brStanja, brSemplova] = size (NNizlaz);
    if viseGresaka == false
        brStanja = 1;
    else
        brStanja = brojStanja;
    end
    duzinaStanja = brSemplova / brStanja;
% sada se izbacuje prvih (n-1) uzoraka, jer ce biti izracunata srednja vrednost prvih n uzoraka i upisana na
% mestu n-tog clana, koji od tog momenta postaje prvi clan
    preradjenaDuzinaStanja = duzinaStanja - (n-1);
    tempPreradjenIzlaz = NNizlaz;
% u sledecoj petlji se za svaku pojedinačnu neispravnost poziva funkcija "preradiStanje", a onda se preradjeno
% stanje pakuje u matricu tempPreradjenIzlaz
    for indeksStanja = 1:brStanja
% Ovde se izdvaja podmatrica koja se odnosi na tekucu neispravnost
        jednoStanje = izdvojStanje (NNizlaz, indeksStanja);
        preradjenoJednoStanje = preradiStanje (jednoStanje, n);
        tempPreradjenIzlaz(:, (indeksStanja - 1)*preradjenaDuzinaStanja + 1 : indeksStanja*preradjenaDuzinaStanja) =
            preradjenoJednoStanje;
    end
    preradjenIzlazCeo = tempPreradjenIzlaz;
    tempPreradjenIzlaz(:, (1+end-(n-1)*brStanja):end) = [];
% u prethodnoj liniji se brisu visak-kolone, koje do ovog momenta ima matrica tempPreradjenIzlaz"
    preradjenIzlaz = tempPreradjenIzlaz;

% Funkcija "preradiStanje" izracunava srednju vrednost uzastopnih n kolona i smesta dobijenu vrednost na mesto
% n-te kolone, a uklanja prvih (n-1) kolona
    function preradjenoStanje = preradiStanje (matricaStanja, n)
        [brRedova, brKolona] = size (matricaStanja);
        tempIzlaz = matricaStanja;
        for i = 1:brRedova
            for j = n:brKolona
                tempIzlaz(i, j) = sum (matricaStanja(i, j-n+1:j)) / n;
            end
        end
        preradjenoStanje = tempIzlaz;
        preradjenoStanje = preradjenoStanje(:, n:end);
    end

% Funkcija "izdvojStanje" iz matrice svih neispravnosti izdvaja one kolone, koje se odnose na neispravnost sa
% indeksom "indexStanja"
    function jednoStanje = izdvojStanje (svaStanja, indexStanja)
        jednoStanje = svaStanja(:, (indexStanja-1)*duzinaStanja + 1:indexStanja*duzinaStanja);
    end
end
end

```

# P4 Izvorni kodovi funkcija napisanih u svrhu vizuelnog prikaza rezultata

## P4.1 *iscrtajNNizlaz* – izvorni kod

```

% iscrtajNNizlaz.m

% Ova funkcija se koristi za vizuelno prikazivanje odziva neuronske mreze
function iscrtajNNizlaz(Matrica,Prag,Naslov,xLabela,yLabela,brojStanja, oznakeStanja,figureCaption)
figureSize=[0 0 0.55 0.7];
polozajIvelicinaOsa=[0.087 0.11 0.88 0.88];
figure('units','normalized','outerposition',figureSize);
set(gcf,'Name',figureCaption,'NumberTitle','off');
h=gca;
set(h,'Position',polozajIvelicinaOsa);
[brRedova,brKolona]=size(Matrica);
intervalVertLinija = brKolona/brojStanja;
OKBoja=[0 0 0]; % crna
FaultIBoja1=[0.8 0 0]; % crvena osnova
FaultIBoja2=[1 0 0.2]; % crvena osnova
FaultIBoja3=[1 0.4 0.4]; % crvena osnova
FaultIBoja4=[1 0.6 0.2]; % crvena osnova
FaultIIBoja1=[0.4 0.6 0]; % zelena osnova
FaultIIBoja2=[0 0.6 0.4]; % zelena osnova
FaultIIBoja3=[0 0.8 0]; % zelena osnova
FaultIIBoja4=[0 0.8 0.8]; % zelena osnova
FaultIIIBoja1=[0 0.2 0.8]; % plava osnova
FaultIIIBoja2=[0.2 0 1]; % plava osnova
FaultIIIBoja3=[0 0.8 1]; % plava osnova
FaultIVBoja1=[1 0 1]; % ljubicasta osnova
FaultIVBoja2=[0.6 0 0.8]; % ljubicasta osnova
boje={OKBoja; FaultIBoja1; FaultIBoja2; FaultIBoja3; FaultIBoja4; FaultIIBoja1; FaultIIBoja2;
      FaultIIBoja3; FaultIIBoja4; FaultIIIBoja1; FaultIIIBoja2; FaultIIIBoja3; FaultIVBoja1; FaultIVBoja2};

OKLine='-';
OKLineWidth=2;
FaultIILine1='-'; % crvena osnova
FaultIILine1Width=2;
FaultIILine2='--'; % crvena osnova
FaultIILine2Width=2;
FaultIILine3='-'; % crvena osnova
FaultIILine3Width=2;
FaultIILine4='--'; % crvena osnova
FaultIILine4Width=2;
FaultIIILine1='-'; % zelena osnova
FaultIIILine1Width=2;
FaultIIILine2='--'; % zelena osnova
FaultIIILine2Width=2;
FaultIIILine3='-'; % zelena osnova
FaultIIILine3Width=2;
FaultIIILine4='--'; % zelena osnova
FaultIIILine4Width=2;
FaultIIILLine1='-'; % plava osnova
FaultIIILLine1Width=2;
FaultIIILLine2='--'; % plava osnova
FaultIIILLine2Width=2;
FaultIIILLine3='-'; % plava osnova
FaultIIILLine3Width=2;
FaultIVLine1='--'; % ljubicasta osnova
FaultIVLine1Width=2;
FaultIVLine2='-'; % ljubicasta osnova
FaultIVLine2Width=2;

linije={OKLine; FaultIILine1; FaultIILine2; FaultIILine3; FaultIILine4; FaultIIILine1; FaultIIILine2; FaultIIILine3;
      FaultIIILine4; FaultIIILLine1; FaultIIILLine2; FaultIIILLine3; FaultIVLine1; FaultIVLine2};

debljineLinija={OKLineWidth; FaultIILine1Width; FaultIILine2Width; FaultIILine3Width; FaultIILine4Width;
      FaultIIILine1Width; FaultIIILine2Width; FaultIIILine3Width; FaultIIILine4Width;
      FaultIIILLine1Width; FaultIIILLine2Width; FaultIIILLine3Width; FaultIIILLine4Width;
      FaultIVLine1Width; FaultIVLine2Width};

loopEnd=brRedova;
for i=1:loopEnd
    boja=boje{i};
    linija=linije{i};
    sirina=debljineLinija(i,1);
    plot(Matrica(i,:), 'Color',boja, 'line',linija, 'LineWidth',sirina);
    hold on;
end

xlim([0 brKolona]);

```

```

yMax=max(max(Matrica));
yMin=min(min(Matrica));
ylim([1.1*yMin 1.5*yMax]);
vrhDijagrama=get(h,'YLim');
yTeksta=0.85*vrhDijagrama(1,2);
fontSize=10;

tekstIndent=15; % ovo je za eksperimentalnu varijantu
% tekstIndent=1; % ovo je za simulacionu varijantu

NN=true; % ovim se odredjuje da li se "oznaciStanja" poziva za dijagram simulacija ili dijagram izlaza
          neuronske mreze
oznaciStanjaBoje(h, intervalVertLinija, brKolona, yTeksta, oznakeStanja, fontSize, tekstIndent, NN, boje);
linijaPraga=Prag*ones(1, length(Matrica(1,:)));
plot(linijaPraga, 'Color', 'k');
title(Naslov, 'FontSize', 12);
xlabel(xLabela, 'FontSize', 12);
ylabel(yLabela, 'FontSize', 12);
set(h, 'XTick', 0: intervalVertLinija: brKolona);
hgexport(gcf, [strrep(figureCaption, '.mat', '') '.png'], hgexport('factorystyle'), 'Format', 'png');
hgsave([strrep(figureCaption, '.mat', '-d') '.fig']);
end

```

## P4.2 oznaciStanjaBoje – izvorni kod

```

% oznaciStanjaBoje.m

% Ova funkcija iscratava vertikalne linije, koje omogucavaju da se jasnije vide pojedina stanja neispravnosti
na vec nacrtanim dijagramima velicina koje se prate (npr. opterecenje motora, pritisak i temperatura u usisnom
kolektoru, broj obrtaja). Osim toga, upisuje i oznake pojedinih stanja na odgovarajucim pozicijama. Parametri
koji se prosledjuju ovoj funkciji su:
% currentAxes - aktuelni koordinatni sistem,
% intervalVertLinija - razmak izmedju susednih vertikalnih linija,
% matricaVrednosti - matrica sa podacima o velicini na cijem dijagramu se oznacavaju stanja,
% brRedova - ukupan broj redova matriceVrednosti tj. ukupan broj semplova u simulacijama u jednoj radnoj
          tacki pomocu fajla MVEMSimStepeniPrenosa.m
% merodavnaKolona - indeks kolone u matriciVrednosti u kojoj se nalazi merodavna velicina
% koeficijentPolozaja - realan broj, pomocu kojeg se lociraju oznake stanja iznad glavne linije dijagrama
% stanja - niz stringova sa oznakama pojedinih stanja
% velicinaSlova - velicina teksta koji na dijagramu oznacava stanja
% tekstIndent - integer koji oznacava pocetak ispisivanja oznake stanja na dijagramu

function oznaciStanjaBoje(currentAxes, intervalVertLinija, brRedova, yPozicijaTeksta,
                          stanja, velicinaSlova, tekstIndent, NN, boje)

hax=currentAxes;
visina=get(hax, 'YLim');
brojac=1;
for i=0: intervalVertLinija: brRedova
    line([i i], visina, 'Color', [0 0 0], 'LineStyle', '-');
    if i < brRedova
        duzinaOznake = length(stanja{brojac});
        switch duzinaOznake
            case 2
                xPozicijaTeksta = tekstIndent+3;
            case 3
                xPozicijaTeksta = tekstIndent+2;
            otherwise
                xPozicijaTeksta = tekstIndent;
        end

        if NN
            bojaTeksta=boje{brojac};
        else
            bojaTeksta=[1 0 0];
            xPozicijaTeksta=xPozicijaTeksta-1;
        end
        % vertikalni polozaj teksta fiksno
        text(i+xPozicijaTeksta, yPozicijaTeksta, stanja(brojac), 'FontSize', velicinaSlova,
            'Color', bojaTeksta, 'FontWeight', 'normal', 'FontName', 'Cambria')
        end
        brojac=brojac+1;
    end
end
end

```

## P4.3 stanjaNN – izvorni kod

```

% stanjaNN.m

% Parametar vrstaSimulacije oznacava da li je motor sa protokomerom ili bez, kao i da li su simulirana
standardna ili nepoznata stanja: 1 (sa protokomerom, standardna stanja); 2 (bez protokomera, standardna
stanja); 3 (sa protokomerom, nepoznata stanja); 4 (bez protokomera, nepoznata stanja)
% Ova funkcija se koristi za obelezavanje stanja posle izlaza iz neuronske mreze.

function stanjaCell = stanjaNN(vrstaSimulacije)
    switch vrstaSimulacije
        case 1 % sa protokomerom, standardna stanja

```

```

tempStanja={['Sve';'OK']; ['MAP';'-10']; ['MAP';'-5']; ['MAP';'+5']; ['MAP';'+10'];
  ['MAF';'90%']; ['MAF';'95%']; ['MAF';'105%']; ['MAF';'110%'];
  ['FaV';'2mm']; ['FaV';'4mm']; ['FaV';'6mm']; ['EGR';'80%']; ['EGR';'60%']];
case 2 % bez protokomera, standardna stanja
tempStanja={['Sve';'OK']; ['IAT';'-15']; ['IAT';'-10']; ['IAT';'+10']; ['IAT';'+15']; ['MAP';'-10'];
  ['MAP';'-5']; ['MAP';'+5']; ['MAP';'+10']; ['FaV';'2mm']; ['FaV';'4mm'];
  ['FaV';'6mm']; ['EGR';'80%']; ['EGR';'60%']];
case 3 % sa protokomerom, nepoznata stanja - za testiranje, varijanta I
tempStanja={['Sve';'OK']; ['Sve';'OK']; ['MAP';'-10']; ['MAP';'-8'];
  ['MAP';'-5']; ['MAP';'-3']; ['MAP';'+5']; ['MAP';'+3'];
  ['MAP';'+10']; ['MAP';'+8']; ['MAF';'90%']; ['MAF';'92%'];
  ['MAF';'95%']; ['MAF';'97%']; ['MAF';'105%']; ['MAF';'103%'];
  ['MAF';'110%']; ['MAF';'108%']; ['FaV';'2mm']; ['FaV';'1mm'];
  ['FaV';'4mm']; ['FaV';'3mm']; ['FaV';'6mm']; ['FaV';'5mm'];
  ['EGR';'80%']; ['EGR';'90%']; ['EGR';'60%']; ['EGR';'70%']];
case 4 % bez protokomera, nepoznata stanja - za testiranje, varijanta I
tempStanja={['Sve';'OK']; ['Sve';'OK']; ['IAT';'-15']; ['IAT';'-13'];
  ['IAT';'-10']; ['IAT';'-6']; ['IAT';'+10']; ['IAT';'+6'];
  ['IAT';'+15']; ['IAT';'+13']; ['MAP';'-10']; ['MAP';'-8'];
  ['MAP';'-5']; ['MAP';'-3']; ['MAP';'+5']; ['MAP';'+3'];
  ['MAP';'+10']; ['MAP';'+8']; ['FaV';'2mm']; ['FaV';'1mm'];
  ['FaV';'4mm']; ['FaV';'3mm']; ['FaV';'6mm']; ['FaV';'5mm'];
  ['EGR';'80%']; ['EGR';'90%']; ['EGR';'60%']; ['EGR';'70%']];
case 5 % sa protokomerom, nepoznata stanja - za testiranje, varijanta II
tempStanja={['Sve';'OK']; ['Sve';'OK']; ['MAP';'-10']; ['MAP';'-12'];
  ['MAP';'-5']; ['MAP';'-7']; ['MAP';'+5']; ['MAP';'+7'];
  ['MAP';'+10']; ['MAP';'+12']; ['MAF';'90%']; ['MAF';'88%'];
  ['MAF';'95%']; ['MAF';'93%']; ['MAF';'105%']; ['MAF';'107%'];
  ['MAF';'110%']; ['MAF';'112%']; ['FaV';'2mm']; ['FaV';'1.7'];
  ['FaV';'4mm']; ['FaV';'3.6']; ['FaV';'6mm']; ['FaV';'5.4'];
  ['EGR';'80%']; ['EGR';'85%']; ['EGR';'60%']; ['EGR';'65%']];
case 6 % bez protokomera, nepoznata stanja - za testiranje, varijanta II
tempStanja={['Sve';'OK']; ['Sve';'OK']; ['IAT';'-15']; ['IAT';'-17'];
  ['IAT';'-10']; ['IAT';'-12']; ['IAT';'+10']; ['IAT';'+12'];
  ['IAT';'+15']; ['IAT';'+17']; ['MAP';'-10']; ['MAP';'-12'];
  ['MAP';'-5']; ['MAP';'-7']; ['MAP';'+5']; ['MAP';'+7'];
  ['MAP';'+10']; ['MAP';'+12']; ['FaV';'2mm']; ['FaV';'1.7'];
  ['FaV';'4mm']; ['FaV';'3.6']; ['FaV';'6mm']; ['FaV';'5.4'];
  ['EGR';'80%']; ['EGR';'85%']; ['EGR';'60%']; ['EGR';'65%']];
case 7 % sa protokomerom, nepoznata stanja - za testiranje, varijanta III
tempStanja={['Sve';'OK']; ['Sve';'OK']; ['MAP';'-10']; ['MAP';'-12'];
  ['MAP';'-5']; ['MAP';'-6']; ['MAP';'+5']; ['MAP';'+6'];
  ['MAP';'+10']; ['MAP';'+12']; ['MAF';'90%']; ['MAF';'88%'];
  ['MAF';'95%']; ['MAF';'94%']; ['MAF';'105%']; ['MAF';'107%'];
  ['MAF';'110%']; ['MAF';'113%']; ['FaV';'2mm']; ['FaV';'1.7'];
  ['FaV';'4mm']; ['FaV';'3.6']; ['FaV';'6mm']; ['FaV';'5.4'];
  ['EGR';'80%']; ['EGR';'85%']; ['EGR';'60%']; ['EGR';'65%']];
case 8 % bez protokomera, nepoznata stanja - za testiranje, varijanta III
tempStanja={['Sve';'OK']; ['Sve';'OK']; ['IAT';'-15']; ['IAT';'-14'];
  ['IAT';'-10']; ['IAT';'-8']; ['IAT';'+10']; ['IAT';'+8'];
  ['IAT';'+15']; ['IAT';'+14']; ['MAP';'-10']; ['MAP';'-12'];
  ['MAP';'-5']; ['MAP';'-6']; ['MAP';'+5']; ['MAP';'+6'];
  ['MAP';'+10']; ['MAP';'+12']; ['FaV';'2mm']; ['FaV';'1.7'];
  ['FaV';'4mm']; ['FaV';'3.6']; ['FaV';'6mm']; ['FaV';'5.4'];
  ['EGR';'80%']; ['EGR';'85%']; ['EGR';'60%']; ['EGR';'65%']];
case 9 % sa protokomerom, nepoznata stanja - za testiranje, varijanta IV
tempStanja={['Sve';'OK']; ['Sve';'OK']; ['MAP';'-10']; ['MAP';'-9'];
  ['MAP';'-5']; ['MAP';'-4']; ['MAP';'+5']; ['MAP';'+4'];
  ['MAP';'+10']; ['MAP';'+9']; ['MAF';'90%']; ['MAF';'92%'];
  ['MAF';'95%']; ['MAF';'96%']; ['MAF';'105%']; ['MAF';'104%'];
  ['MAF';'110%']; ['MAF';'108%']; ['FaV';'2mm']; ['FaV';'1.4'];
  ['FaV';'4mm']; ['FaV';'4.3']; ['FaV';'6mm']; ['FaV';'5.7'];
  ['EGR';'80%']; ['EGR';'73%']; ['EGR';'60%']; ['EGR';'57%']];
case 10 % bez protokomera, nepoznata stanja - za testiranje, varijanta IV
tempStanja={['Sve';'OK']; ['Sve';'OK']; ['IAT';'-15']; ['IAT';'-14'];
  ['IAT';'-10']; ['IAT';'-8']; ['IAT';'+10']; ['IAT';'+8'];
  ['IAT';'+15']; ['IAT';'+14']; ['MAP';'-10']; ['MAP';'-9'];
  ['MAP';'-5']; ['MAP';'-4']; ['MAP';'+5']; ['MAP';'+4'];
  ['MAP';'+10']; ['MAP';'+9']; ['FaV';'2mm']; ['FaV';'1.4'];
  ['FaV';'4mm']; ['FaV';'4.3']; ['FaV';'6mm']; ['FaV';'5.7'];
  ['EGR';'80%']; ['EGR';'73%']; ['EGR';'60%']; ['EGR';'57%']];
end
stanjaCell=tempStanja;
end

```

# P5 Izvorni kodovi programa napisanih u okviru razvoja laboratorijskog dijagnostičkog modela

## P5.1 radneTackeMotora – izvorni kod

```

% radneTackeMotora.m

% Ovo je program za određivanje radnih tacaka motora na kojima ce se vrsiti glavna merenja u eksperimentalnom
delu rada. To bi trebalo da budu tacke koje ce priblizno pokriti radni dijapazon motora u kojem ce on
najcesce raditi pri kretanju jednog prosecnog vozila po ravnoj podlozi.

clear; n = [1200:2:3700]; % dijapazon brojeva obrtaja u kojem se odredjuju radne tacke

% Ucitavanje podataka o motoru iz eksel fajla. Ovaj fajl treba da bude formatiran tako da u prvoj koloni budu
brojevi obrtaja na kojima su snimane parcijalne karakteristike motora, a u drugoj, trecoj itd. podaci o
odgovarajucim obrtnim momentima motora.
[imeFajla, putanjaFajla] = uigetfile( ...
    {'*.xls','Excell Files (*.xls)'},...
    'Izaberite fajl sa parcijalnim brzinskim karakteristikama');
if imeFajla~=0
    parcKarMomenta=xlsread([putanjaFajla imeFajla]);
else
    parcKarMomenta=0;
    errordlg('Ne moze se izvršiti određivanje radnih tacaka motora, ako ne ucitate njegove parcijalne
    karakteristike','Greska!','modal');
    return;
end

[redovi kolone] = size(parcKarMomenta);
brojParcKar = kolone-1; % ovde se određuje broj snimljenih parcijalnih karakteristika
brojRadnihTacaka = redovi; % broj radnih tacaka u okviru jedne parcijalne karakteristike

% Sada se vrsi fitovanje snimljenih tacaka u parcijalne brzinske karakteristike momenta motora
% Iscrtavanje fitovanih parcijalnih karakteristika obrtnog momenta motora
bojaMotora=[0.6 0.2 0]; % zelena osnova
bojaOtpora=[0 0 1]; % plava osnova
bojaParcijalnihTacaka=[0 0.8 0]; bojaRadnihTacaka=[1 0 0];
figureSize=[0 0 0.6 0.85]; polozejIvelicinaOsa=[0.087 0.11 0.88 0.82];
debljinaKrive=1;
xLabela='Boj obrtaja [1/min]';
yLabela='Moment motora (Moment otpora) [Nm]';
figure('Name',imeFajla,'NumberTitle','off','units','normalized','outerposition',figureSize);
set(gca,'Position',polozajIvelicinaOsa);
for i = 1:brojParcKar
    p(:,i) = polyfit(parcKarMomenta(:,1),parcKarMomenta(:,i+1),2);
    Mfit(:,i) = polyval(p(:,i),n);
    plot(n,Mfit(:,i),'LineWidth',debljinaKrive,'Color',bojaMotora); % bez ucrtanih tacaka merenja parcijalnih
    karakteristika momenta motora

    xlim([1200 3600]); ylim([0 80]);
    hold on;
end

% Podaci potrebni za sracunavanje karakteristike momenta otpora na zamajcu
etaTR = 0.9; % koeficijent iskoriscenja transmisije [-]
iGP = 3.867; % prenosni odnos glavnog prenosnika [-]
iM = [3.727 2.053 1.32 0.967 0.794]; % prenosni odnosi menjaca u pojedinim stepenima prenosa [-]
rD = 0.3; % dinamicki radijus tocka [m]
mVozila = 1000; % masa vozila [kg]
A = 2; % ceona površina vozila [m2]
cW = 0.35; % koeficijent otpora vazduha
nagibPuti = 0; % nagib puta [%]
brojStepeniPrenosa = length(iM);
% Izracunavanje karakteristika momenata otpora po pojedinim stepenima prenosa
for j = 1:brojStepeniPrenosa
    vVozila(:,j) = 0.377*rD*n/(iM(j)*iGP); % brzina vozila [km/h]
% Određivanje sila otpora [N]
Fw(:,j) = 0.0473*cW*A*vVozila(:,j).^2; % otpor vazduha
Ff(:,j) = mVozila*9.81*(0.01+(5.42e-6)*vVozila(:,j)+(1.05e-11)*vVozila(:,j).^4); % kotrljanje
Fnagiba(:,j) = mVozila*9.81*nagibPuti/100; % nagib puta
Fo(:,j) = Ff(:,j)+Fw(:,j)+Fnagiba(:,j); % ukupna sila otpora kretanja vozila [N]
Mo(:,j) = Fo(:,j)*rD/(etaTR*iGP*iM(j)); % ukupni moment otpora na zamajcu motora [Nm]
% Iscrtavanje brzinskih karakteristika momenta otpora
plot(n,Mo(:,j),'Color',bojaOtpora,'LineWidth',debljinaKrive);
    hold on;
end

xlabel(xLabela,'FontName','Cambria','FontSize',14);
ylabel(yLabela,'FontName','Cambria','FontSize',14);
% Određivanje presečnih tacaka parcijalnih brzinskih karakteristika obrtnog momenta motora i brzinskih
karakteristika momenta otpora u pojedinim stepenima prenosa

```

```

for k = 1:brojStepeniPrenosa
    for m = 1:brojParcKar
        [nPreseka,Mpreseka]=curveintersect(n,Mfit(:,m),n,Mo(:,k));
        plot(nPreseka,Mpreseka,'ro','LineWidth',1,'MarkerEdgeColor','k','MarkerFaceColor',
            bojaRadnihTacaka,'MarkerSize',6);
        if isempty([nPreseka,Mpreseka])==0
            rpm(m,k)=nPreseka(1,1);
            M(m,k)=Mpreseka(1,1);
        end
    end
end
grid;
fileNameOffered='Radne tacke motora.xls';
[filename, pathname, filterindex] = uiputfile( {'*.xls','Excell Files (*.xls)'},...
    'Snimi radne tacke kao excell fajl',fileNameOffered);

if filename~=0
    xlswrite([pathname filename],rpm,'Sheet1','B3');
    xlswrite([pathname filename],M,'Sheet1','I3');
    set(gcf,'Name',filename);
    hgexport(gcf, [strrep(filename,'.mat','') '.png'],hgexport('factorystyle'),'Format','png');
    hgsave([strrep(filename,'.mat','-d') '.fig']);
else
    errordlg('Podaci nisu snimljeni!','Upozorenje','modal');
end
end

```

## P5.2 grupisaneNeispravnosti - izvorni kod

```

% grupisaneNeispravnosti.m

```

```

% Ova funkcija generise jedan excel fajl koji objedinjuje excel fajlove sa podacima iz eksperimenata
% sprovedenih sa razlicitim neispravnostima za jednu radnu tacu motora.

```

```

function matricaStanja=grupisaneNeispravnosti(otvorLeptira,test)
    filenames=uipickfiles('FilterSpec','*.xls');
    if ~isempty(filenames)
        if iscell(filenames) % Ako je filename Cell of arrays, izvodi se procedura za multiselect
            brojStanja=length(filenames);
            Temp=cell(3,1);
            Temp{3,1}=[];
            Prefiks=['Lep' num2str(otvorLeptira) ' '];
            if test==true
                Prefiks=['Test' Prefiks];
            end
            imeFajlaKumulativno='';

            for i=1:brojStanja
                putanjaIime=filenames{i};
                [putanja imeFajla]=razdvojPutanjuIime(putanjaIime);
                fileRead=xlswrite([putanja imeFajla]);
                imeBezEkstenzije=strrep(imeFajla,'.xls','');
                Temp{i,1}=fileRead;
                imeFajlaKumulativno=[imeFajlaKumulativno '-' skратиIмеFajla(imeBezEkstenzije)];
            end
            matricaStanja=cell2mat(Temp);
        end
        fileNameOffered=[Prefiks imeFajlaKumulativno '.xls'];
        [filename, pathname, filterindex] = uiputfile( ...
            {'*.xls','Excell Files (*.xls)'},...
            'Snimi rezultate simulacije kao excell fajl',fileNameOffered);
        if filename~=0
            xlswrite([pathname filename],matricaStanja);
        end
    end
    % Crtanje dijagrama za poznata stanja
    iscrtajDijagramExp(pathname, filename, matricaStanja, brojStanja, stanjaExp(2));
    else
        errordlg('Podaci nisu snimljeni!','Upozorenje','modal');
        return;
    end
end
errordlg('Niste izabrali fajlove. Operacija se ne moze izvrstiti.','Greska!','modal');
return;
end

function skracenoIme=skратиIмеFajla(originalName)
    svePovlake=strfind(originalName,'-');
    poloZajPrvePovlake=svePovlake(1);
    osnovniDeoImena=originalName(strfind(originalName,'Mom')+5:poloZajPrvePovlake);
    korekcijeDeoImena=originalName(strfind(originalName,'RPM'):strfind(originalName,'Red')-3);
    korekcijeDeoImenaSkracen=strrep(korekcijeDeoImena,'RPM','');
    korekcijeDeoImenaSkracen=strrep(korekcijeDeoImenaSkracen,'MAP','');
    korekcijeDeoImenaSkracen=strrep(korekcijeDeoImenaSkracen,'IAT','');
    korekcijeDeoImenaSkracen=strrep(korekcijeDeoImenaSkracen,'PW','');
    skracenoIme=[osnovniDeoImena korekcijeDeoImenaSkracen];
end
end

```