



UNIVERZITET U NIŠU
MAŠINSKI FAKULTET



Emina (Pavle) Petrović

**RAZVOJ HIJERARHIJSKE STRUKTURE
UPRAVLJANJA MOBILNIM ROBOTOM ZA
PRAĆENJE LJUDI NA BAZI ROBUSNE STEREO
ROBOTSKE VIZIJE
DOKTORSKA DISERTACIJA**

Niš, 2017.



UNIVERSITY OF NIŠ
FACULTY OF MECHANICAL
ENGINEERING



Emina (Pavle) Petrović

**DEVELOPMENT OF HIERARCHICAL CONTROL
STRUCTURE OF A MOBILE ROBOT FOR
HUMAN TRACKING BASED ON ROBUST
ROBOTIC STEREO VISION**

DOCTORAL DISSERTATION

Niš, 2017

Podaci o doktorskoj disertaciji

- Mentor:** Prof.dr Vlastimir Nikolić, red. prof., Univerzitet u Nišu, Mašinski fakultet
- Naslov:** Razvoj hijerarhijske strukture upravljanja mobilnim robotom za praćenje ljudi na bazi robusne stereo robotske vizije
- Rezime:** Predmet istraživanja disertacije odnosi se na razvoj i primenu hijerarhijske strukture upravljanja u kojoj se na višem nivou upravljanja izvršavaju algoritmi koji primenom stohastičkih metoda u robotskoj viziji detektuju ljude, estimiraju njihovu poziciju, prate ih i prepoznaju njihovo ponašanje, u cilju izvršavanja zadataka u kojima se robot ponaša kao čovekov saradnik.
- U ovoj tezi su ponuđena neka rešenja koja predstavljaju korak napred ka rešavanju problema s kojima se sreće sistem robotske vizije koji treba da obezbedi pouzdane ulaze u upravljački modul mobilnog robota-pratioca čoveka. Razvijen je robusan modul vizije, za praćenje ljudi koji može biti primenjen u različitim aplikacijama u kojima je potrebno da robot radi zajedno sa čovekom i koji može da se primeni na različite tipove mobilnih robota. Posebna pažnja u disertaciji je posvećena je integraciji, testiranju i eksperimentalnoj verifikacija stohastičkih algoritma za praćenje ljudi, kao što su Kalmanovi i Partikle filteri, kao i uporedna analiza algoritama za rešavanje problema robotskog praćenja ljudi.
- Deo istraživanja predstavljen u ovoj tezi se oslanja na zajednički naučni rad, koje su izveli istraživači sa Mašinskog fakulteta Univerziteta u Nišu i istraživači sa Instituta za automatiku (IAT) Univerziteta u Bremenu. Modul stereo vizije za detekciju ljudi razvijen na Institutu za automatiku Univerziteta u Bremenu (IAT), korišćen je u testiranju modula praćenja koji je razvijen u ovoj tezi. Pored ovog sistema detekcije korišćeni su, takođe i sistemi za detekciju ljudi koji koriste 3D senzore, kao što je Asus Xtion PRO LIVE 3D senzor.
- Fokus teze je na razvoju simulacionog okruženja i njegovog upravljačkog sistema, kao i razvoj modula za praćenje, estimaciju i prepoznavanje ponašanja čoveka, koje predstavlja podršku razvoju i primeni stvarnog (eng. real-world) upravljačkog sistema. Razvijeno simulaciono okruženje može lako biti korišćeno i za druge mobilne robote uz odgovarajuće modifikacije. Razvijeni algoritmi su evaluirani na Mašinskom fakultetu u Nišu, u okviru ove doktorske disertacije, gde je implementirano napredno hijerarhijsko upravljanje mobilnim robotom DaNI, firme National Instruments, korišćenjem 3D senzora ASUS Xtion PRO LIVE koji u eksperimentalnom laboratorijskom scenariju predstavlja senzor robotske vizije za modul prepoznavanja i praćenja ljudi. Takođe, na IAT-u, implementiran je modul vizije koji

se sastoji od dva sub-modula, modula stereo vizije za detekciju čoveka i modula praćenja zasnovanim na Kalmanovom filteru razvijenom u okviru ove doktorske teze.

| | |
|-------------------------------------|---|
| Naučna oblast: | Tehničko – tehnološke nauke |
| Naučna disciplina: | Upravljanje sistemima i robotika |
| Ključne reči: | Hijerarhijsko upravljanje, robusna robotska vizija , Kalmanov filter, Partikle filter, mobilni robot, stereo vizija |
| UDK: | 007.52:519.712 (043.3) 621-52:004.896 (043.3) |
| CERIF klasifikacija: | T 125 Automatizacija, robotika, kontrolni inženjering |
| Tip licence Kreativne zajednice: | CC BY-NC-ND |

Data on Doctoral Dissertation

Doctoral Supervisor: Prof.dr Vlastimir Nikolić, red. prof., Univerisity of Nišu, Faculty of Mechanical Engineering

Title: Development of hierarchical control structure of a mobile robot for human tracking based on robust robotic stereo vision

The main topic of this doctoral thesis refers to the development and implementation of the hierarchical control structure in which the algorithms are being executed on the high-level control. By applying the stochastic methods in robotic vision to these algorithms, we can detect people, estimate their position, follow them and recognise their actions in order to carry out tasks where the robot behaves like a human's collaborator. In this thesis, some solutions are offered and present a step forward towards solving the problems that robotic vision system that provides reliable inputs to the control module of the mobile human-collaboration robot, is facing. The robust vision module for human tracking which can be applied in various applications where is necessary for robots to work together with humans and which can be applied on different types of mobile robots, was developed.

Abstract: In this thesis, it is devoted a special attention to the integration, testing and experimental verification of the stochastic algorithms for human tracking such as Kaman and Particle filters, as well as a comparative analysis of algorithms for solving problems of robotic people tracking. A part of the research presented in this thesis is based on a scientific collaboration between the researchers from the Faculty of Mechanical Engineering of Nis and the researchers from the Institute of Automation (IAT) of the University of Bremen.

The stereo vision module for a human detection that was developed at the Institute of Automatic Control, University of Bremen (IAT), was used for testing the tracking module which was developed in this thesis. Beside the detection system, the systems for human detection that use 3D sensors, such as the Asus Xtion PRO LIVE 3D sensor were used. The main focus of this thesis was the development of a simulation environment and its control system, as well as the development of modules for human tracking, estimation of a human position and recognition of a human behavior. The simulation environment represents the support to the development and implementation of the real world control system. By adding the appropriate modifications, other mobile robots can easily use this simulation environment. The developed algorithms are evaluated on Faculty of Mechanical Engineering, University of Niš as a part of the doctoral dissertation. In this dissertation, advanced hierarchical

control was implemented for the purpose of controlling the mobile robot DaNI, developed by the company National Instruments. This advanced hierarchical control was implemented by using 3D sensor Asus Xtion Pro Live which in the laboratory experimental scenario represents the robotic vision sensor for the detection modules and human tracking. In addition, at the IAT, the vision module which consists of two sub-modules was implemented. These two sub-modules are the stereo vision for a human detection and the tracking module based on the Kalman filter developed in this doctoral thesis.

| | |
|--------------------------------|--|
| Scientific Field: | Technical – technological sciences |
| Scientific Discipline: | Control systems and Robotics |
| Key Words: | Hierarchical control, robust robotic vision, kalman filter, Particle filter, mobile robot, stereo vision |
| UDC: | 007.52:519.712 (043.3) 621-52:004.896 (043.3) |
| CERIF Classification: | P125 Automation, robotics, control engineering |
| Creative Commons License Type: | CC BY-NC-ND |

Zahvalnica

Želim da se zahvalim svom mentoru prof. dr Vlastimiru Nikoliću na ogromnoj podršci tokom izrade disertacije. Veliku zahvalnost dugujem van. prof. dr Danijeli Ristić-Durrant koja me je za sve vreme istraživanja i izrade ove teze vodila i usmeravala, kao i prof. dr Predragu Rajkoviću za nesebičnu pomoć i savetima. Veliku zahvalnost dugujem i kolegi sa Instituta za Automatiku Univerziteta u Bremenu Adrianu Leuu, kao i kolegama sa Mašinskog fakulteta Univerziteta u Nišu Miši Tomiću, dr Milošu Simonoviću, dr Ivanu Ćiriću, Neveni Tomić i Veljku Begoviću na saradnji i podršci tokom izrade ove teze.

Takođe zahvaljujem mom vereniku Miroljubu na ogromnoj podršci i neizmernom strpljenju tokom istraživanja i samog pisanja ove teze.

PODACI O MENTORU I ČLANOVIMA KOMISIJE

Mentor

dr Vlastimir Nikolić, redovni profesor
Univerzitet u Nišu, Mašinski fakultet u Nišu
UNO: Automatsko upravljanje i robotika

Članovi komisije za ocenu i odbranu doktorske disertacije

dr Danijela Ristić-Durrant, vanredni profesor
Univerzitet u Nišu, Mašinski fakultet u Nišu
UNO: Automatsko upravljanje i robotika

dr Dragan Antić, redovni profesor
Univerzitet u Nišu, Elektronski fakultet u Nišu
UNO: Automatika

dr Žarko Čojbašić, redovni profesor
Univerzitet u Nišu, Mašinski fakultet u Nišu
UNO: Automatsko upravljanje i robotika

dr Predrag Rajković, redovni profesor
Univerzitet u Nišu, Mašinski fakultet u Nišu
UNO: Matematika i informatika

Datum odbrane doktorske disertacije

SADRŽAJ

| | |
|---|-----|
| 1. Uvod | 1 |
| 1.1. Pregled dosadašnjih istraživanja..... | 2 |
| 1.2. Glavni doprinosi ove teze | 4 |
| 1.3. Opis sistema..... | 5 |
| 1.4. Organizacija teze | 9 |
| 2. Detekcija čoveka zasnovana na stereo viziji | 10 |
| 2.1. Akvizicija slika..... | 10 |
| 2.2. Izračunavanje mape dispariteta | 12 |
| 2.3. Segmentacija mape dispariteta | 13 |
| 2.4. Klasifikacija objekata na osnovu karakteristika | 14 |
| 2.4.1. 2D karakteristike | 14 |
| 2.4.2. 3D karakteristike | 14 |
| 2.4.3. Klasifikacija objekata i detekcija čoveka | 15 |
| 3. Modeliranje i simulacija mobilnog robota sa diferencijalnim pogonom..... | 17 |
| 3.1. Kinematsko modeliranje mobilnog robota sa diferencijalnim pogonom | 19 |
| 3.1.1. Model direktne kinematike..... | 20 |
| 3.1.2. Neholonomska ograničenja kretanja robota..... | 22 |
| 3.2. Dinamičko modeliranje mobilnog robota sa diferencijalnim pogonom..... | 25 |
| 3.2.1. Lagranževov metod dinamike | 26 |
| 3.2.1.1. Redukovan dinamički model | 32 |
| 3.2.3. Njutm-Ojlerov metod dinamike | 34 |
| 3.4. Simulacija sistema | 40 |
| 4. Projektovanje kontrolera za praćenje čoveka | 45 |
| 4.1. Razvoj hijerarhijske strukture upravljanja mobilnim robotom sa diferencijalnim pogonom za praćenje ljudi | 46 |
| 4.2. Upravljački sistem niskog nivoa –kontroler brzine..... | 47 |
| 4.3. Upravljački sistem srednjeg nivoa..... | 50 |
| 4.3.1. Ljapunovljeve analize stabilnosti | 52 |
| 4.3.2. Simulacija pozicionog kontrolera..... | 53 |
| 4.3.2.1. Testiranje upravljačke šeme za konstantu brzinu čoveka i željeno rastojanje $d=0$ m ... | 56 |
| 4.3.2.2. Testiranje upravljačke šeme za konstantu brzinu čoveka i željeno rastojanje $d=1.5$ m . | 63 |
| 4.3.2.3. Testiranje upravljačke šeme za promenljivu brzinu čoveka i željeno rastojanje $d=0$... | 68 |
| 4.4. Optimizacija parametara multivarijabilnog PD kontrolera rojevima čestica | 70 |
| 5. Stohastički algoritmi za rešavanje problema praćenja ljudi na osnovu podataka dobijenim senzorom robotske vizije | 74 |
| 5.1. Bajesovi filteri – teorijska osnova | 76 |
| 5.1.1. Kalmanov filter | 78 |
| 5.1.2. Prošireni Kalmanov filter | 82 |
| 5.1.3. Partikle filter..... | 87 |
| 5.1.3.1. Monte Karlo | 88 |
| 5.1.3.2. Uzorkovanje po značajnosti..... | 89 |
| 5.1.3.3. Sekvencijalno uzorkovanje po značajnosti –SIS metod | 92 |
| 5.1.3.4. Sekvencijalno uzorkovanje sa resamplovanjem-SIR metod..... | 94 |
| 6. Eksperimentalni rezultati praćenja čoveka na osnovu podataka dobijenim senzorom robotske vizije..... | 97 |
| 6.1. Mobilna robotska platforma Neobotix MP-M470 sa integrisanim sistemom stereo vizije..... | 97 |
| 6.1.1. Inicijalizacija Kalmanovog filtera..... | 99 |
| 6.1.2. Rezultati praćenja čoveka na osnovu Kalmanovog filtera | 102 |

| | |
|--|-----|
| 6.2. Mobilna robotska platforma DaNI sa ASUS Xtion 3D senzorom | 104 |
| 6.2.1 Rezultati praćenja čoveka na osnovu Kalmanovog filtera | 108 |
| 6.2.2. Rezultati praćenja čoveka na osnovu proširenog Kalmanovog filtera | 110 |
| 6.2.3. Rezultati praćenja čoveka na osnovu Partikle filtera | 114 |
| 6.3. Uporedna analiza estimatora | 117 |
| 6.4. Simulacija sistema sa modulom praćenja zasnovanim na Kalmanovom filteru | 120 |
| 7. Upravljanje mobilnim robotom na osnovu informacija sa visokog nivoa upravljanja..... | 123 |
| 7.1. Model neuronske mreže za klasifikaciju u cilju prepoznavanja ponašanja čoveka | 124 |
| 7.2. Modul odlučivanja..... | 127 |
| 7.3. Integracija DaNI mobilnog robota sa sistemom robotkse vizije | 129 |
| 7.4. praćenje čoveka mobilnom robotskom platformom DaNI sa ASUS Xtion 3D senzorom..... | 134 |
| 8. Zaključci i dalja istraživanja..... | 138 |
| LITERATURA | 140 |
| Dodatak A | 150 |
| Dodatak B | 151 |

1. UVOD

Sposobnost praćenja čoveka mobilnim robotom je jedna od esecijalnih karakteristika modernih uslužnih robotskih sistema, od kojih se očekuje da rade sa ljudima i dele radni prostor s njima. Ovi roboti moraju biti u stanju da detektuju poziciju čoveka u prostoru i prate ga. Ovakva sposobnost nudi mogućnost primene robota u različitim aplikacijama kao što su bezbednosni roboti koji detektuju i prate provalnike, roboti koji ostvaruju interakciju sa čovekom i servisni roboti koji prate osobu pružajući kontinualnu asistenciju, kao što je na primer asistencija starih osoba u obavljanju svakodnevnih delatnosti. Praćenje ljudi je poseban izazov za postizanje pune autonomije robota.

Mobilni roboti koji prate čoveka su opremljeni odgovarajućim sensorima koji obezbeđuju ulaznu informaciju za sistem upravljanja robotom. U velikom broju slučajeva, kamera kao senzor vizije je osnovni senzor za dobijanje informacije o objektima nad kojima je potrebno izvršiti određeni zadatak u robotskim sistemima. Tako se sistem robotske vizije u poslednje vreme javlja kao esencijalni sastavni deo robotskih sistema i predstavlja osnovni izvor informacija na osnovu koga robotski sistem vrši percepciju okoline. Praćenje ljudi sistemom robotske vizije je problem detekcije osobe iz jedne u drugu sliku u nizu slika. Da bi se izgradio sistem robotske vizije koji je robustan u odnosu na različite spoljašnje uticaje kao što su promena osvetljenja, promena ugla snimanja i prisustvo mnogo različitih objekata u sceni, potrebno je izvršiti predikciju pozicije čoveka koji se prati mobilnim robotom. Robot kontinualno procenjuje poziciju i brzinu čoveka u lokalnom robotskom kordinatnom sistemu da bi adekvatno upravljao svojim kretanjem.

Za robote koji pružaju podršku osobama u izvršavanju raznih zadataka kao što su roboti koji pomažu osobama sa invaliditetom, roboti-pratioci starijih ljudi ili mobilni roboti koji asistiraju ljudima u inspekciji kontaminiranih/opasnih okruženja, ključni uslov za generisanje adekvatnog ulaza u upravljački sistem, pored detekcije i praćenja ljudi, je prepoznavanje i predikcija ponašanja ljudi. Da bi izvršili ove zadatke roboti moraju da poseduju sposobnosti slične kognitivnim sposobnostima ljudi za rezonovanje, zaključivanje, planiranje. Izvršavanje ovih kognitivnih akcija podrazumeva intezivno izvršavanje matematičkih algoritama što zbog svoje sporosti mora da bude na visokom nivou strukture upravljanja (*eng. high-level control*) u odnosu na takozvano upravljanje mobilnom platformom na srednjem (*eng. medium-level control*) i niskom nivou (*eng. low-level control*) koje se izvršava u realnom vremenu.

Praćenje ljudskog tela i pokreta kao i njihova analiza dobila je značajnu pažnju u kompjuterskoj viziji u protekloj deceniji. Ovo je uglavnom motivisano željom razumevanja ljudskih gestova i poza za izgradnju nove generacije korisničkog interfejsa. Međutim, iako je mnogo toga urađeno

prethodnih godina, veliki broj problema je i dalje nerešen. Ti problemi se prvenstveno odnose na nedovoljnu robustnost sistema vizije u odnosu na brojne spoljašnje poremećaje kao što su promenljivo osvetljenje u sceni koja se snima i „zagađenost“ slike kamere različitim šumovima.

1.1. PREGLED DOSADAŠNJIH ISTRAŽIVANJA

U protekloj deceniji urađeno je dosta u oblasti detekcije i praćenja ljudi sa ciljem postizanja autonomije mobilnih robota korišćenjem različitih senzora u cilju prikupljanja informacija o okruženju i robotskoj reakciji na promene u neposrednoj okolini. Kompjuterska vizija predstavlja jedan od najpopularnijih senzora percepcije u cilju postizanja autonomnosti mobilnih sistema. Razvoj upravljačkih struktura za upravljanje mobilnim robotima u cilju praćenja na bazi vizije predstavlja aktivnu oblast istraživanja.

Da bi postigli bolje performanse praćenja čovaka u radu [1] sistem stereo vizije je korišćen u kombinaciji sa laserom za merenje udaljenosti (*eng. Laser Range Finder (LRF)*). Takođe neki od istraživača su razvijali metode koje uključuju odelo sa sensorima [2] pored korišćenja sistema vizije na robotu, međutim primena ovakve metoda je neadekvatna u većini aplikacija, zbog potrebe da čovek nosi specijalno odelo.

Poslednjih decenija urađeno je dosta u oblasti detekcije i praćenja ekstrahovanjem različitih karakteristika sa slike (*eng. image features*) kao i primenom različitih metoda klasifikacije. [3],[4],[5],[6],[7].

Stereo kamera je veoma popularna u detekciji i praćenju pokretnih objekata, zato što se stereo vizijom dobija informacija o dubini scene. U svojoj tezi Leu [8] je razvio pouzdane algoritme za detekciju i praćenje ljudi u realnom vremenu zasnovanih na stereo viziji u zatvorenom prostoru (*eng. indoor*) kao i u otvorenom (*eng. outdoor*). Beymer i Konolige [9] su razvili metod za konstantno praćenje ljudi detektujući osobu koristeći podatke o udaljenosti čoveka od stacionarne stereo kamere.

Zadnjih godina razvijeni su algoritmi za detekciju ljudi i praćenje zasnovanih na viziji koji omogućavaju on-line upotrebu 3D senzora sa stuktuiranom svetlošću kao što su Microsoft Kinect [10] i Asus Xtion PRO LIVE [11]. Ove kamere daju informaciju o dubini scene zajedno sa slikom scene u boji.

Biblioteke otvorenog koda (*eng. open-source libraries*) kao što je OpenNI [12] koriste dodatne informacije koje pružaju ove kamere da bi se omogućila on-line detekcija i praćenje čoveka.

Upravljanje neholonomskim mobilnim robotima na bazi vizije je rađeno u [13],[14], dok je u [15] prezentovana studija vizualne navigacije mobilnih robota u zatvorenom i otvorenom okruženju.

U oblasti kompjuterske vizije, praćenje pokretnog objekta mobilnim robotom je relativno težak problem [69]. Upotreba fazi logike je veoma česta kod upravljanja mobilnih robota [70], kao i neuro-fazi pristup koji koristi tehnike mašinskog učenja za podešavanje parametara funkcija pripadnosti [71][72]. Takođe, korišćene su i veštačke potencijalne funkcije [73], kao i histogrami vektorskih polja [74] za navigaciju mobilnim robotom. Darrell i drugi [75] integrišu informacije o dubini (*eng. depth information*), segmentaciju po bojama (*eng. color segmentation*) i detekciju lica (*eng. face detection*) za praćenje određene osobe. Feyerer i Zell [76] predlažu algoritam za praćenje koji je zasnovan na stereo viziji i na podacima dobijenih laserom. Generalno, vizualno praćenje možemo podeliti prema različitim kriterijumima. Jedan od kriterijuma može biti broj praćenih objekata/osoba (jedan objekat/osoba, više objekata/osoba, grupa objekata/osoba), zatim sama karakteristika osobe koju pratimo, npr. možemo da pratimo telo osobe, ili samo pojedine delove tela npr. njeno lice, zatim ruke, noge itd. Zatim možemo koristiti stacionarnu ili pokretnu kameru. Često se podela praćenja zasnovanom na viziji vrši i prema metodama koje se koriste: praćenje na osnovu modela (*eng. model-based tracking*), praćenje na osnovu regiona (*eng. region-based tracking*), praćenje na osnovu aktivne konture (*eng. active contour-based tracking*) i praćenje na osnovu karakteristika (*eng. feature-based tracking*) [77].

Veliki broj metoda praćenja za određivanje lokacije čoveka može se podeliti i na determinističke i probalističke metode praćenja. Od determinističkih metoda u vizualnom praćenju najčešće se koristi MeanShift [78][79]. MeanShift pretpostavlja da je ponašanje mete nepromenljivo u toku kretanja. Kada stereo vizija ne daje rezultate usled poremećaja u sistemu, često ovaj metod daje loše rezultate. Probalističke metode zapraćenje ljudi su zasnovane na bazi Bajesovih filtera, kao što su Kalmanov filter [80][81][82][83] i Sekvencijalni Monte Karlo estimator poznat kao Partikle filter [84][85] koji je mnogo robustniji i samim tim pogodniji za primenu u realnom slučaju.

U radu [19] modifikovani Kalmanov filter se koristi da bi smanjili tzv. region od interesa (region u kome je najveća verovatnoća pojavljivanja detektovanog čoveka) u cilju smanjivanja vremena potrebnog za segmentaciju, jer se ne segmentira cela slika, nego samo region od interesa. U radu [116] koriste se standardni Kalmanov filter za praćenje na osnovu informacija dobijenih laserom. Kim i drugi [117] su dizajnirali estimator stanja u na osnovu Kalmanovog filtera da bi izbegli nepouzadane podatke dobijene kamerom. Da bi upravljali mobilnim robotom u cilju praćenja pokretnog objekta, pri čemu se robot kreće najkraćom putanjom, estimirane su linearna i ugaona brzina pokretnog objekta.

Satake i Miura [3] koriste prošireni Kalmanov filter za praćenje čoveka mobilnim robotom u dinamičnom okruženju. Robot u svakom trenutku procenjuje poziciju i brzinu čoveka u robotskom koordinatnom sistemu, koje se koriste za adekvatno upravljanje mobilnim robotom.

Choi i Kim [77] koriste Partikle filter za praćenje 3D pozicije elipsoidnog modela glave čoveka, koji je efikasan pri velikim rotacijama i translacijama. Schulz i drugi [118] su predstavili robot, opremljen sa dva lasera, jedan za scenu ispred robota, a drugi za scenu iza robota. Na osnovu kombinacije Partikle filtera i probalističkog filtera sa podacima povezanim zajedničkom verovatnoćom (*eng.* Joint probabilistic data-association filter) ovaj robot je u mogućnosti da prati više osoba. Bellotto i Hu [119] su predstavili efikasno rešenje za praćenje i identifikovanje ljudi mobilnim robotom koristeći multisenzorsku fuziju podataka. Sistem koristi laserski uređaj za detekciju nogu čoveka i PTZ (*eng.* pan-tilt-zoom camera) kameru za nalaženje lica osoba, a zatim se ovi podaci šalju u sekvencijalni nederivacijski Kalmanov filter (*eng.* sequential Unscented Kalman filter) u cilju praćenja više osoba u realnom vremenu. U radu [120] Bellotto i Hu su takođe dali i poređenje Kalmanovog, nederivacijskog Kalmanovog i Partikle filtera, za praćenje više osoba korišćenjem fuzije različitih senzora.

Za robote koji pružaju podršku osobama u izvršavanju raznih zadataka, ključni uslov je generisanje adekvatnog ulaza u upravljački sistem. Praćenje ljudi sistemom robotske vizije je problem praćenja kretanja osobe iz jedne u drugu sliku u nizu slika (frejmova) i veoma često se pokretni objekti na segmentiranoj slici ograničavaju graničnim pravougaonim okvirom, a centar pravougaonika se predstavlja kao centar mase objekta koji pratimo [23][86]. Za izgradnju sistema za praćenje, zasnovanim na viziji, koji je robusan u odnosu na promenu osvetljenja, promenu ugla snimanja i prisustvo mnogo različitih objekata u sceni, moramo da odgovorimo na veoma važan problem: predviđanje lokacije osobe koja se prati na sledećem frejmu. Izbor najboljeg filtera za korišćenje zavisi od više faktora, među kojima su najznačajniji: linearnost/nelinearnost sistema, verovatnoća distribucije, neizvesnosti i poslednje, ali ne i najmanje značajno – kompjuterska efikasnost, budući da robot kontinuirano procenjuje poziciju i brzinu čoveka u lokalnom robotskom koordinatnom sistemu kako bi adekvatno upravljao svojim kretanjem.

1.2. GLAVNI DOPRINOSI OVE TEZE

U ovoj tezi su ponuđena neka rešenja koja predstavljaju korak napred ka rešavanju problema s kojima se sreće sistem robotske vizije koji treba da obezbedi pouzdane ulaze u upravljački modul mobilnog robota-pratioca čoveka. Deo istraživanja predstavljen u ovoj tezi se oslanja na zajednički naučni rad, koje su izveli istraživači sa Mašinskog fakulteta Univerziteta u Nišu i istraživači sa Instituta za automatiku (IAT) Univerziteta u Bremenu u okviru dva bilateralna projekta, “Novi pristup detekciji i praćenju ljudi u robotici” i “Robusno prepoznavanje pokreta u cilju dostizanja sinergije čoveka i robota”. Modul stereo vizije za detekciju ljudi [16],[17],[18], razvijen na Institutu za automatiku Univerziteta u Bremenu (IAT), korišćen je u testiranju modula praćenja koji je

razvijen u ovoj tezi. Pored ovog sistema detekcije korišćeni su, takođe i sistemi za detekciju ljudi koji koriste 3D senzore, kao što su Microsoft Kinect i Asus Xtion.

Osnovni ciljevi i doprinosi u ovoj tezi su:

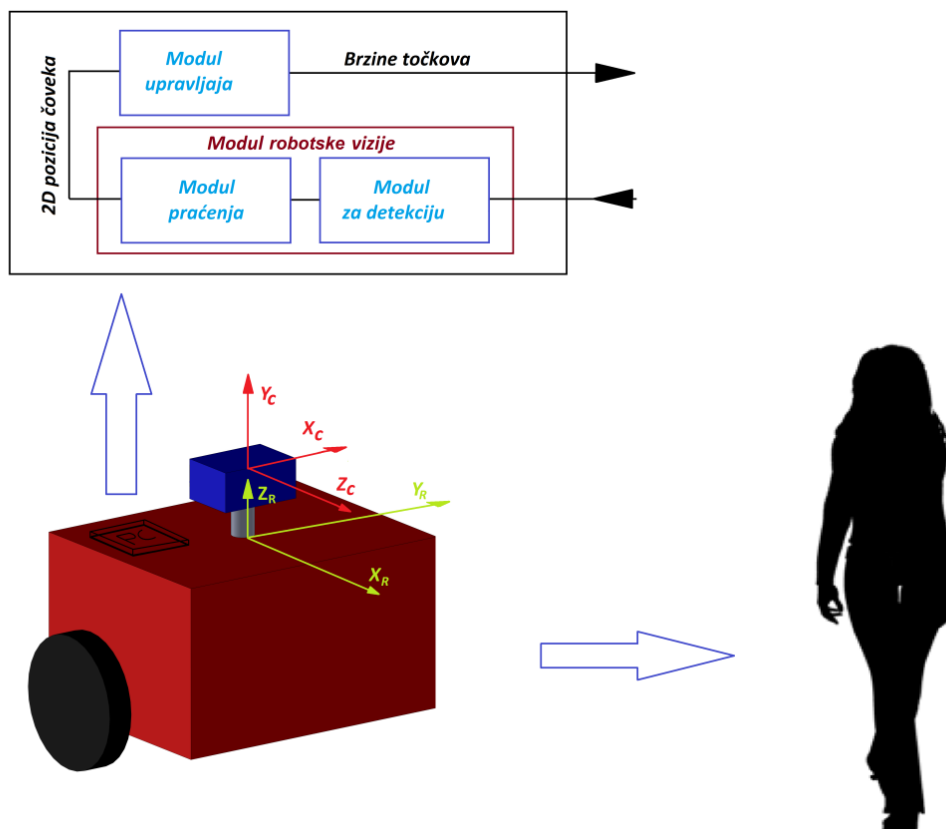
- Razvoj robusnog modula vizije, koji je esencijalan modul kod sistema upravljanja mobilnim robotom za praćenje ljudi u različitim aplikacijama u kojima je potrebno da robot radi zajedno sa čovekom i koji može da se primeni na različite tipove mobilnih robota.
- Integracija, testiranje i eksperimentalna verifikacija stohastičkih algoritma za praćenje ljudi na osnovu Bajesovih filtera, kao što su Kalmanovi i Partikle filteri, kao i uporedna analiza algoritama za rešavanje problema robotskog praćenja ljudi.
- Razvoj modula za prepoznavanje ponašanja čoveka.
- Razvoj hijerarhijske strukture upravljanja koja na visokom novou vrši praćenje i prepoznavanje ponašanja čoveka i generisanje ulaza za niže nivoe upravljanja u cilju izvršavanja zadataka u kojima se robot ponaša kao čovekov saradnik.
- Razvoj upravljačkih algoritama srednjeg i nižeg nivoa za upravljanje mobilnim robotom na osnovu informacija dobijenih sa visokog hijerarhijskog nivoa upravljanja.
- Razvoj simulacionog okruženja koje predstavlja podršku razvoju i primeni stvarnog (*eng. real-world*) upravljačkog sistema.
- Razvijeno simulaciono okruženje može lako biti korišćeno i za druge mobilne robote uz odgovarajuće modifikacije.
- Razvijeni algoritmi su evaluirani na Mašinskom fakultetu u Nišu, u okviru ove doktorske disertacije, gde je implementirano napredno hijerarhijsko upravljanje mobilnim robotom DaNI, firme National Instruments, korišćenjem 3D senzora ASUS Xtion PRO LIVE koji u eksperimentalnom laboratorijskom scenariju predstavlja senzor robotske vizije za modul prepoznavanja i praćenja ljudi.
- Takođe, na IAT-u, implementiran je modul vizije koji se sastoji od dva sub-modula, modula stereo vizije za detekciju čoveka i modula praćenja zasnovanim na Kalmanovom filteru razvijenom u okviru ove doktorske teze. [19]

1.3. OPIS SISTEMA

Sistem robotske vizije razvijen, implementiran i evaluiran u okviru ove teze treba da obezbedi ulaze za upravljanje mobilnim robotom koji treba da radi sa čovekom i pomaže mu prilikom obavljanja različitih zadataka tako što će ga pratiti i služiti za odlaganje određenih predmeta na mobilnu platformu. Predstavljeni robotski sistem upravljan na bazi robotske vizije treba da bude

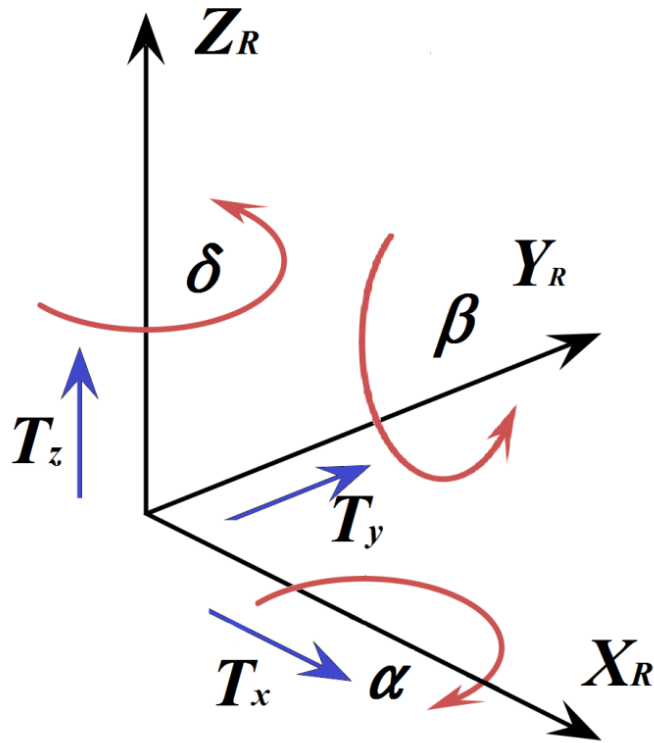
sposoban da detektuje i prati ljude u slikama dobijenim robotskom vizijom i da rezultate praćenja koristi za upravljanje mobilnim robotom tako da robot prati čoveka u radnom scenariju održavajući konstantno rastojanje od čoveka. Takođe, robot treba da ima mogućnost da “predvidi” nameru čoveka da priđe robotu i da u tom slučaju stane i dopusti čoveku da odloži predmet na mobilnu robotsku platformu.

Blok dijagram predloženog robotskog sistema za praćenje čoveka zasnovanog na robotskoj viziji je prikazan na slici 1.1.



Slika 1.1: Blok šema robotskog sistema za praćenje čoveka

Budući da senzor robotske vizije daje podatke o poziciji čoveka u odnosu na koordinatni sistem kamere $\{C\}$, a upravljanje mobilnim robotom se vrši na osnovu pozicije čoveka u robotskom koordinatnom sistemu $\{R\}$, potrebno je izvršiti konverziju pozicije čoveka iz koordinatnog sistema kamere u koordinatni sistem robota. Odnos između koordinatnog sistema kamere i koordinatnog sistema robota dat je jednačinom (1.1). Osnovna ideja je da se ovi koordinatni sistemi mogu povezati jednostavnim koordinatnim transformacijama, kao što su rotacija (R) i translacija (T). Osnovna ideja translacije i rotacije je prikazana na slici 1.2.



Slika 1.2: Osnovna ideja translacije i rotacije

Uglovi α, β, δ predstavljaju rotacije osa koordinatnog sistema kamere oko osa x, y, z u robotskom koordinatnom sistemu, dok T_x, T_y i T_z predstavljaju translacije duž osa x, y, z u robotskom koordinatnom sistemu.

$$\begin{bmatrix} x_R \\ y_R \\ z_R \end{bmatrix} = \mathbf{R} \begin{bmatrix} x_c \\ y_c \\ z_c \end{bmatrix} + \mathbf{T} \quad (1.1)$$

$$\mathbf{R} = \mathbf{R}_x \mathbf{R}_y \mathbf{R}_z \quad (1.2)$$

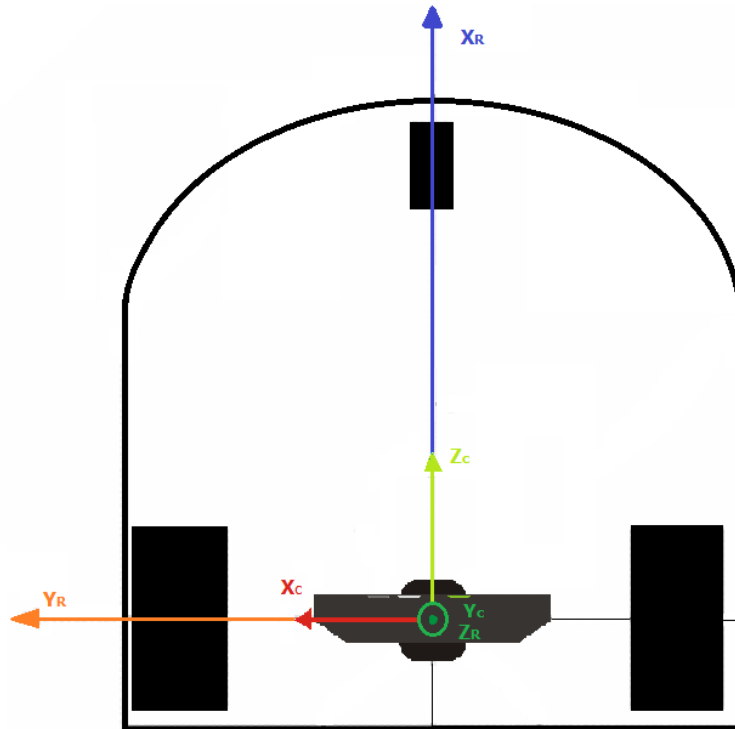
$$\mathbf{R}_x = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(\alpha) & -\sin(\alpha) \\ 0 & \sin(\alpha) & \cos(\alpha) \end{bmatrix} \quad (1.3)$$

$$\mathbf{R}_y = \begin{bmatrix} \cos(\beta) & 0 & \sin(\beta) \\ 0 & 1 & 0 \\ -\sin(\beta) & 0 & \cos(\beta) \end{bmatrix} \quad (1.4)$$

$$\mathbf{R}_z = \begin{bmatrix} \cos(\delta) & -\sin(\delta) & 0 \\ \sin(\delta) & \cos(\delta) & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (1.5)$$

$$\mathbf{T} = \begin{bmatrix} T_x \\ T_y \\ T_z \end{bmatrix} \quad (1.6)$$

U našem sistemu koji je prikazan na slici 1.3, kamera se nalazi na samom robotu i orijentisana je tako da su ose X_R i Z_c , Y_R i X_c , Z_R i Y_c paralelne, tj. uglovi rotacije iznose $\alpha=0^\circ$, $\beta=90^\circ$ i $\delta=90^\circ$.



Slika 1.3: Veza između koordinatnog sistema kamere {C} i koordinatnog sistema robota {R}

Takođe je pretpostavljeno da su translacija T_x i T_y jednake nuli, dok translacija duž ose Z_R isnosi $T_z = 0.4$ m. (Vrednost translacije $T_z = 0.4$ se odnosi na robotsku platformu sa DaNI robotom i Asus Xtion PRO LIVE 3D senzorom, dok za mobilni robotski sistem IAT-ea sa stereo kamerom T_z je oko 1 m). Odnos između koordinatnog sistema kamere i koordinatnog sistema robota je konacno dat jednačinom (1.7).

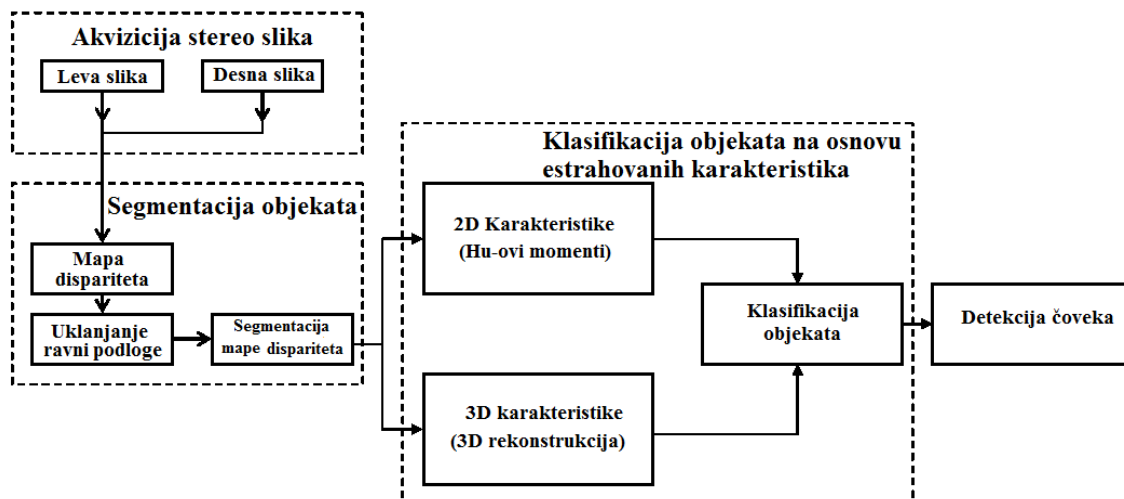
$$\begin{bmatrix} x_R \\ y_R \\ z_R \end{bmatrix} = \begin{bmatrix} 0 & 0 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} x_c \\ y_c \\ z_c \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ T_z \end{bmatrix} \quad (1.7)$$

1.4. ORGANIZACIJA TEZE

Nakon uvoda i pregleda literature datog u ovom poglavlju, u poglavlju 2 dat je detaljan opis sistema za detekciju čoveka zasnovanog na stereo viziji. Modeliranje i simulacija mobilnog robota sa diferencijalnim pogonom su predstavljenu poglavlju 3. Hijerarhijska upravljačka struktura, kao i detaljna analiza upravljačkih algoritama srednjeg i niskog nivoa upravljanja, kao i simulacioni rezultati testiranja ovih upravljačkih algoritama su predstavljeni u poglavlju 4. U poglavlju 5, dat je opis Bajesovih filtera za praćenje čoveka na osnovu informacija dobijenih senzorom robotske vizije, dok su u poglavlju 6 dati eksperimentalni rezultati praćenja čoveka na osnovu informacija dobijenih stereo vizijom i Asus Xtion PRO LIVE 3D senzorom robotske vizije. Poglavlje 7 je posvećeno razvoju klasifikatora za prepoznavanje ponašanja čoveka i modulu za odlučivanje na visokom hijerarhijskom nivou upravljanja, kao i opisu korišćene realne mobilne platforme i implementaciji predložene upravljačke stukture. U poglavlju 8 su dati zaključci i pravci daljeg istraživanja.

2. DETEKCIJA ČOVEKA ZASNOVANA NA STEREO VIZIJI

Blok dijagram sistema za detekciju ljudi na bazi stereo vizije koji je predstavljen u radu [19], a koji zajedno sa modulom praćenja razvijenim u ovoj tezi čini predloženi modul robotske vizije je prikazan na slici 2.1. Individualni moduli sistema za detekciju su objšnjeni u sledećoj sekciji.

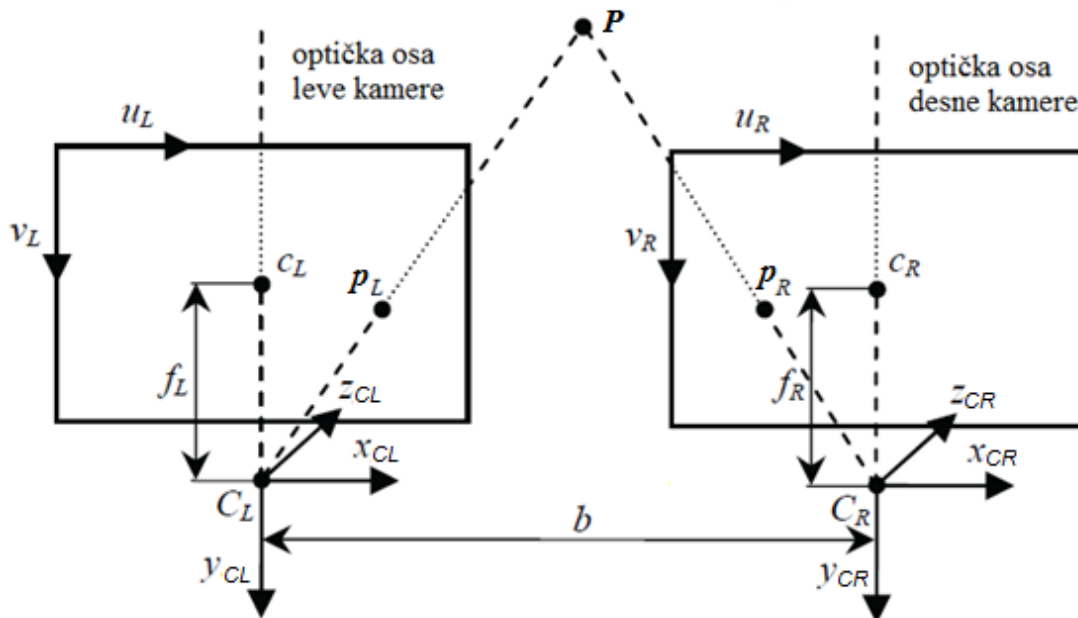


Slika 2.1: Blok dijagram sistema za detekciju ljudi zasnovanog na stereo viziji

2.1. AKVIZICIJA SLIKA

Algoritmi za prepoznavanje ljudi, koji su zasnovani na viziji podrazumevaju procesiranje slika koje su snimljene digitalnom kamerom [8]. Tokom akvizicije slike svetlost ulazi kroz objektiv kamere i fokusira se na senzor slike. Vreme ekspozicije utiče na osvetljenost slike, jer duže vreme ekspozicije omogućava da više svetla padne na senzor, što čini sliku svetlijom. Ostali važni parametri akvizicije slike su veličina blende i osetljivost senzora na svetlost.

Sistem stereo kamere sadrži dve pojedinačne kamere koje su postavljene tako da su njihove optičke ose uglavnom paralelne kao što se može videti na slici 2.2. Ukoliko ose kamera nisu paralelne, onda se nakon akvizicije slike vrši tzv. rektifikacija tako da se slike tretiraju kao da su dobijene paralelnim kamerama [23].



Slika 2.2: Sistem stereo kamere sastavljen od leve i desne kamere; stereo slike sa koordinatnim sistemom slika i koordinatnih sistema kamera [20]

Za svaku pojedinačnu kameru može se definisati koordinatni sistem slike (u, v) , kao i koordinatni sistem same kamere (x, y, z) . Ova dva koordinatna sistema grupišu parametre kamere u dve grupe parametara: unutrašnji (*eng. intrinsic*) (žižna daljina f , bazna linija (*eng. base line*) i centar kamere c (*eng. principal point*) i spoljašnji (*eng. extrinsic*) (rotacija i translacija kamera) [20].

Ovi parametri kamere određuju način na koji se tačka iz trodimenzionalnog (3D) prostora projektuje na ravan slike.

Prilikom procesa akvizicije svaka tačka iz 3D prostora koja 'leži' u vidnom polju stereo kamere, projektuje se na ravni slika obe pojedinačne kamere. 3D lokacija tačke koje ispunjava ovaj uslov može se povratiti (rekonstruirati) ako je poznat lokacija projekcija u obema slikama, zajedno sa parametrima stereo kamere.

2.2. IZRAČUNAVANJE MAPE DISPARITETA

Stereo vid bazira se na pronalaženju podudarnosti između korespodentnih tačaka na slici leve i desne kamere. Ukoliko su optičke ose ovih dveju kamera paralelene, tzv. epipolarne linije biće paralele po horizontali, čime se pojednostavljuje proces nalaženja korespodentnih tačaka [21].

Uz poznati razmak između kamera, tj. baznu liniju (eng. baseline), moguće je izračunati koordinate tačaka u 3D prostoru. U cilju izracunavanja mape dispariteta, uobičajno je da se slika leve stereo kamere smatra referentnom dok je desna slika korespodentna slika.

Izračunavanje mape dispariteta je zasnovano na rešavanju problema korespodencije i izračunanju razlika u koordinatama korespodentnih tačaka u stereo slikama [22].

Za tačku u prostoru sa koordinatama $P(x_c, y_c, z_c)$, korespodentne tačke u levoj i desnoj slici su $p_L(u_L, v_L)$ i $p_R(u_R, v_R)$. Na osnovu koordinata korespodentnih tačaka disparitet je:

$$d_s = u_L - u_R \quad (2.1)$$

d_s : vrednost dispariteta.

u_L, u_R : koordinate korespodentnih tačaka u levoj i desnoj slici respektivno.

Na slici 2.2 je prikazan par stereo slika i mapa dispariteta dobijena na osnovu tih slika.



Slika 2.3: Stereo par i mapa dispariteta [23]

Mapa dispariteta je siva slika (eng. gray-scale), u kojoj tamniji pikseli pripadaju udaljenijim objektima, dok svetliji pikseli ogovaraju objektima koji su bliži kameri. Koordinate piksela na mapi dispariteta korespodentne su koordinatama na levoj stereo slici, budući da se obično uzima leva stereo slika kao referentna prilikom pravljenja mape dispariteta.

Kada sračunamo mapu dispariteta, na osnovu sličnih trouglova (triangulacijom) može se odrediti udaljenost do tačke $P(x_c, y_c, z_c)$ objekta u prostoru ili dubina objekta (z_c) [21].

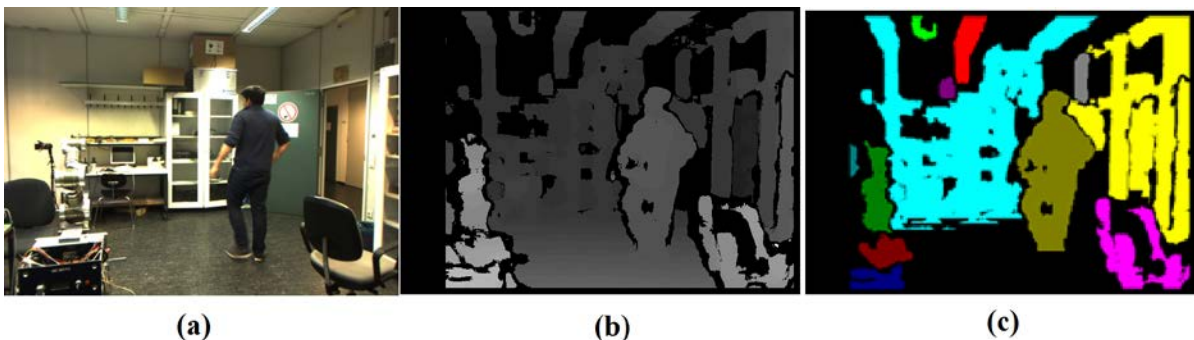
$$z_c = \frac{b \cdot f}{d_s} \quad (2.2)$$

$$x_c = \frac{b \cdot u_L}{d_s} \quad (2.3)$$

$$y_c = \frac{b \cdot v_L}{d_s} \quad (2.4)$$

2.3. SEGMENTACIJA MAPE DISPARITETA

Svrha segmentacije je da pojednostavi mapu dispariteta, podeli je u podregione ili setove piksela kako bi sadržaj bio razumljiviji i jednostavniji za dalju računarsku analizu [8]. Segmentacija se vrši tako što se grupišu pikseli koji imaju istu ili veoma blisku vrednost i pretpostavlja se da pripadaju istom objektu [19]. Naime, susedni pikseli u mapi dipariteta koji pripadaju jednom objektu, imaju sličnu vrednost dispariteta, dok na ivicama objekta, razlika u vrednostima dispariteta objekta i pozadine je velika. Ovi prelazi u vrednostima dispariteta su korišćeni za proces segmentacije u metodu datom u [24].



Slika 2.4: a) Leva stereo slika čoveka koji hoda ispred robota, b) Mapa dispariteta, c) Segmentirana mapa dispariteta [19]

U scenariju, koji se razmatra u ovoj tezi, tj. scenariju kada mobilni robot prati čoveka, rezultati primene metoda segmentacije prikazani su na slici 2.4.

Različito obojeni regioni na slici 2.4.c) reprezentuju segmentirane objekte koji se nalaze na različitim odstojanjima od stereo kamere koja se nalazi na mobilnom robotu. Kao što se može videti na slici, pod je uklonjena iz segmentirane slike (predstavljeno crnom bojom), kako bi se izbeglo spajanje poda sa drugim objektima, uključujući i čoveka, koji stoji na tlu. Uklanjanje ravni poda je urađeno detektovanjem regiona u donjim delovima mape dispariteta, čije su vrednosti dispariteta postepeno menjaju, tj. čiji se gradijent u vertikalnom pravcu postepeno menja. Za razliku od piksela u ravni podloge, pikseli u mapi dispariteta koji pripadaju regionima objekata postavljenim vertikalno u odnosu na pod imaju skoro konstantne vrednosti dispariteta.

2.4. KLASIFIKACIJA OBJEKATA NA OSNOVU KARAKTERISTIKA

Posle izvršene segmentacije mape dispariteta, izračunavaju se različite karakteristike koje opisuju segmentirane regione. Izabrane karakteristike su definisane tako da se omogući jasna razlika između ljudi i ostalih objekata u robotskom okruženju. Karakteristike koje se koriste mogu se podeliti u dve grupe: 2D i 3D karakteristike.

2.4.1. 2D karakteristike

Da bi se opisali oblici svakog segmentiranog regiona, koriste se takozvani invarijantni Hu-ovi momenti [25], kao dokazano efikasni deskriptori oblika. Oni se sračunavaju uz pomoć piksela svakog segmentiranog regiona prema sledećim formulama:

$$H_1 = \frac{\mu_{20}}{\mu_{00}^2} + \frac{\mu_{02}}{\mu_{00}^2} \quad (2.5)$$

$$H_2 = \left(\frac{\mu_{20}}{\mu_{00}^2} - \frac{\mu_{02}}{\mu_{00}^2} \right)^2 + \frac{4\mu_{11}}{\mu_{00}^2} \quad (2.6)$$

$$H_3 = \left(\frac{\mu_{30}}{\mu_{00}^{5/2}} - 3 \frac{\mu_{12}}{\mu_{00}^{5/2}} \right)^2 + \left(3 \frac{\mu_{21}}{\mu_{00}^{5/2}} - \frac{\mu_{03}}{\mu_{00}^{5/2}} \right)^2 \quad (2.7)$$

gde je μ_{pq} centralni moment definisan kao:

$$\mu_{pq} = \sum_u \sum_v (u - \bar{u}_c)^p (v - \bar{v}_c)^q I(u, v), \quad p, q = 0, 1, 2, \dots \quad (2.8)$$

\bar{u}_c i \bar{v}_c : predstavljaju koordinate centra mase segmentiranog objekta u koordinatnom sistemu slike, koji predstavljaju srednje vrednosti u i v koordinata svih piksela iz segmentiranog regiona.

$I(u, v)$: predstavlja nivo intenziteta tačke na slici (vrednost piksela) sa koordinatama u i v .

U slučaju segmentirane slike, $I(u, v)$ je 1 za svaki segmentirani piksel koji pripada objektu od interesa i 0 za svaki piksel koji pripada pozadini (nesegmentiran piksel).

2.4.2. 3D karakteristike

Da bi se efikasno razlikovao čoveka od drugih predmeta u okruženju, Hu-ovi momenti, kao deskriptori oblika segmentiranih objekata u 2D segmentiranoj mapi dispariteta, su podržani 3D

karakteristikama, kao što su stvarna (3D) visina i širina objekta. Za izračunavanje ovih dveju karakteristika, potrebno je prvo da se granični okvir svakog segmentiranog objekta definiše kao najmanji pravougaonik koji sadrži segmentiran region na 2D segmentiranoj mapi dispariteta. Zatim se identifikuju gornji levi (LU) i donji desni ugao (RB) svakog graničnog okvira i njihove koordinate u segmentiranoj mapi dispariteta (u_{LU}, v_{LU}) i (u_{RB}, v_{RB}). Da bi se izračunale stvarna 3D širina i visina objekta, vrši se 3D rekonstrukcija ili "2D to 3D mapping" prema sledećim formulama:

$$x_{LU} = \frac{b \cdot u_{LU}}{d_s}, y_{LU} = \frac{b \cdot v_{LU}}{d_s}, z_{LU} = \frac{b \cdot f}{d_s} \quad (2.9)$$

$$x_{RB} = \frac{b \cdot u_{RB}}{d_s}, y_{RB} = \frac{b \cdot v_{RB}}{d_s}, z_{RB} = \frac{b \cdot f}{d_s} \quad (2.10)$$

gde je:

f : žižna daljina kamere (u pikselima)

b : bazna linija, tj, rastojanje između centara stereo kamere (eng. baseline)

d_s : vrednost dispariteta (vrednost piksela) odgovarajuće tačke u mapi dispariteta .

Sa izračunatim 3D koordinatama tačaka koje odgovaraju tačkama na uglovima graničnog okvira segmentiranog objekta, visina h i širinu w objekta se sračunavaju prema sledećim formulama:

$$h = Y_{LU} - Y_{RB}, \quad (2.11)$$

$$w = X_{RB} - X_{LU} \quad (2.12)$$

Ove karakteristike se koriste u sistemu, koji se razmatra u ovoj tezi, za klasifikaciju objekta da bi smo odredili da li je segmeniran objekat je čovek ili ne [23].

2.4.3. Klasifikacija objekata i detekcija čoveka

Problem klasifikacije se javlja kada je potrebno da objekat dodelimo unapred definisanoj grupi ili klasi na osnovu karakteristika koje opisuju taj objekat. Tradicionalne statističke metode klasifikacije kao što je diskriminantna analiza su zasnovane na Bajesovoj teoriji odlučivanja. Kod ovih metoda, polazni model verovatnoće se pretpostavlja tako da izračunava posteriornu verovatnoću na osnovu koje se donosi odluka. Efikasnost ovih metoda u velikoj meri zavisi od raznih pretpostavki ili uslova pod kojim se razvijaju modeli, što je jedno od glavnih ograničenja ovih metoda. Da bi uspešno primenili ove modele, moramo prethodno dobro poznavati karakteristike i mogućnosti samih modela [26].

Zadnjih godina, neuronske mreže su se pojavile kao važan alat kod klasifikacije objekata i obećavajuća alternativa konvencionalnim metodama klasifikacije. Prednost neuronskih mreža leži u sledećem: neuronske mreže su samoadaptivni metodi i mogu da se prilagode podacima bez ikakve eksplicitne specifikacije funkcionalne ili distributivne forme za osnovne modele. Druga prednost neuronskih mreža je u tome da su univerzalni funkcionalni aproksimatori koji mogu aproksimirati bilo koju funkciju sa proizvoljnom tačnošću [27]. Takođe, neuronske mreže su nelinerani modeli, što ih čini fleksibilnim u modelovanju kompleksnih modela.

Za klasifikaciju objekta u prikazanom sistemu detekcije razvijenom na IAT korišćenaje neuronska mreža sa povratnom propagacijom (eng.Backpropagation Neural Network) sa jednim skrivenim slojem [27].

Za obuku kalsifikatora, tj. podešavanje parametra neuronske mreže, korišćeno je 577 setova podataka za karakteristike (H_1, H_2, H_3, h, w) , koje su dobijene iz segmentirane mape dispariteta, a 423 novih setova podataka je korišćeno za validaciju i testiranje.

Jednom kada je čovek uspešno klasifikovan, može se odediti njegova 3D poziciju u odnosu na koordinatni sistem leve stereo kamere na osnovu 3D koordinata centra mase segmetiranog regiona $(\bar{u}_{HC}, \bar{v}_{HC})$, koji je klasifikovan kao čovek, prema sledećim formulama:

$$z_{HC} = \frac{b \cdot f}{d_s} \quad (2.13)$$

$$x_{HC} = \frac{b \cdot \bar{u}_{HC}}{d_s} \quad (2.14)$$

$$y_{HC} = \frac{b \cdot \bar{v}_{HC}}{d_s} \quad (2.15)$$

Rezultati klasifikacije su se pokazali veoma dobrim, sa tačnoću od 97%. Do pogrešne klasifikacije je dolazilo u slučajevima kada objekti iz okoline zaklanjaju čoveka ili kada prilikom segmentiranja čovek bude u kontaktu sa objektima iz okoline.

Sve ovo ukazuje na potrebu za integrisanje modula kojim će se predvideti/estimirati položaj čoveka i na taj način obezbediti adekvatne ulaze u modul za upravljanje mobilnim robotom, čak i u slučaju neuspeha sistema za detekciju čoveka.

Modul za predikciju/estimaciju položaja čoveka, biće opisan u jednom od narednih odeljaka u tezi.

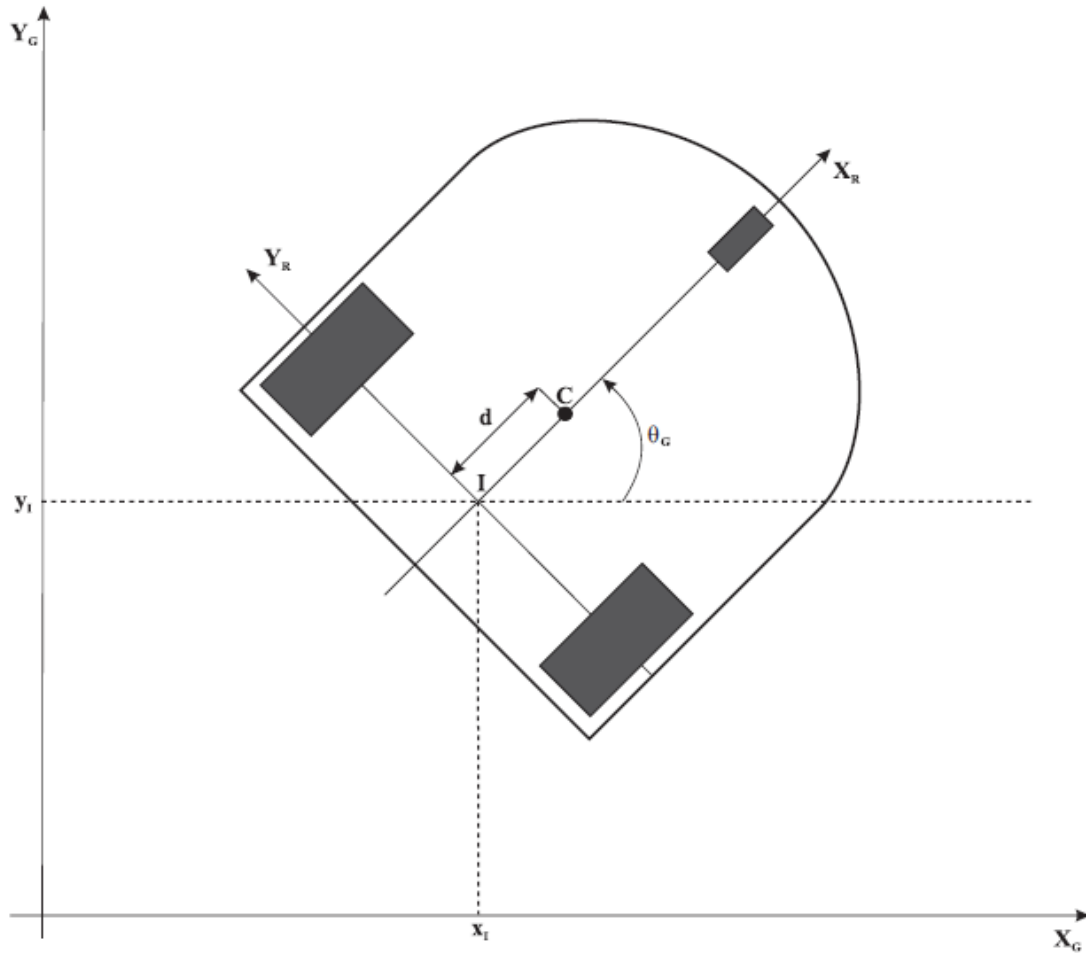
3. MODELIRANJE I SIMULACIJA MOBILNOG ROBOTA SA DIFERENCIJALNIM POGONOM

Mobilni roboti imaju mogućnost da se „kreću“ u svom okruženju i nisu fizički fiksirani za jednu lokaciju. Mobilni roboti se mogu podeliti na *holonomske* i *neholonomske*. U slučaju diferencijalnog pogona mobilnog robota, koji spada u neholonomske mobilne robote, ovo znači da robot može da se kreće samo u nekim pravcima (napred i nazad), ali ne u svim (levo ili desno) [39]. Ova teza razmatra samo neholonomske mobilne robote sa diferencijalnim pogonom. Za projektovanje, razvoj, modifikaciju i upravljanje mobilnog robotskog sistema potrebno je razumevanje tog sistema kao i odgovarajući prikaz istog, to jest neophodan je takozvani „model“ sistema. Model predstavlja idealizaciju stvarnog sistema. Za modeliranje komponenata robotskog sistema potrebno je koristiti analogne procedure. Na ovakav način modeli komponenata se adekvatno integrišu kako bi se dobio celokupan model [39]. Modeliranje platforme diferencijalnog pogona mobilnog robota sastoji se iz kinematskog i dinamičkog modeliranja. Kinematsko modeliranje se bavi geometrijskim vezama u sistemu kao i matematikom kretanja zanemarujući sile koje deluju na sistem. Za razliku od kinematskog, dinamičko modeliranje proučava kretanja kod kojih su sile i energija uključeni u model sistema [39]. Svaki deo modeliranja razmatranog robotskog sistema je zasebno objašnjen u ovom poglavlju, na sličan način kao i u radu [39]. Dobijeni precizni model je onda korišćen za simulaciju kretanja mobilnog robota korišćenjem Matlab/Simulink softvera. Razvijeni simulacioni model je zatim korišćen za testiranje predložene hijerarhijske strukture upravljanja, kao što će biti pokazano u poglavlju 4.

Prvi korak kod mehaničkog modeliranja je definisanje odgovarajućih koordinatnih sistema za odgovarajuću mobilnu platformu [39]. Osnovna funkcija koordinatnih sistema je da se prikaže pozicija robota. Sledeća dva koordinatna sistema se koriste za modeliranje i kontrolu mobilnog robota:

- Globalni (referentni) sistem: $\{X_G, Y_G\}$ - globalni (referentni) koordinatni sistem.
- Robotski sistem: $\{X_R, Y_R\}$ - koordinatni sistem zakačen za robota.

Ova dva koordinatna sistema prikazana su na Slici 3.1:



Slika 3.1: Globalni i robotski koordinatni sistem

Uvođenje ovih koordinatnih sistema pomaže kod kinematskog modeliranja robota o čemu će biti reči u sledećem odeljku. Neophodno je objasniti mapiranje između ova dva sistema. Pozicija bilo koje tačke na robotu u globalnom i robotskom sistemu može biti definisana na sledeći način:

$$\mathbf{q}_G = [x_G \quad y_G \quad \theta_G]^T \quad (3.1)$$

$$\mathbf{q}_R = [x_R \quad y_R \quad \theta_R]^T \quad (3.2)$$

gde su x_G , y_G i θ_G koordinate i ugao rotacije neke tačke na robotu u globalnom koordinatnom sistemu, a x_R , y_R i θ_R koordinate i ugao rotacije te iste tačke u robotskom koordinatnom sistemu.

Mapiranje ova dva sistema se vrši pomoću standardne ortogonalne rotacione transformacije:

$$\dot{\mathbf{q}}_R = \mathbf{R}(\theta)\dot{\mathbf{q}}_G \quad (3.3)$$

$$\mathbf{R}(\theta) = \begin{bmatrix} \cos\theta & \sin\theta & 0 \\ -\sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (3.4)$$

pri čemu je ugao rotacije robota u globalnom koordinatnom sistemu označen bez subskripta G , $\theta = \theta_G$.

Korišćenjem jednačina iznad stvara se veza između brzine robota u lokalnom i brzine robota u globalnom sistemu što je jako bitno u kinematici robota.

3.1. KINEMATSKO MODELIRANJE MOBILNOG ROBOTA SA DIFERENCIJALNIM POGONOM

Cilj kinematskog modeliranja robota je izračunavanje brzine robota u globalnom sistemu kao funkciju brzine točkova i geometrijskih parametara robota (konfiguracija koordinata)[39]. Drugim rečima treba sračunati brzinu robota $\dot{\mathbf{q}}_G = [\dot{x}_G \dot{y}_G \dot{\theta}_G]^T$, kao funkciju brzine točkova $\dot{\phi}_R$ i $\dot{\phi}_L$ i geometrijskih parametara robota ili kao vezu između kontrolnih parametara ($\dot{\phi}_R$ i $\dot{\phi}_L$) i ponašanja sistema u prostoru stanje. Kinematika robota obično ima dve glavne analize, jedna je direktna kinematika, a druga je inverzna kinematika:

- Direktna kinematika:

$$\dot{\mathbf{q}}_G = \begin{bmatrix} \dot{x}_G \\ \dot{y}_G \\ \dot{\theta}_G \end{bmatrix} = f(\dot{\phi}_R, \dot{\phi}_L, \text{geometrijskiparametri}) \quad (3.5)$$

- Inverzna kinematika:

$$\begin{bmatrix} \dot{\phi}_R \\ \dot{\phi}_L \end{bmatrix} = f(\dot{x}_G, \dot{y}_G, \dot{\theta}_G) \quad (3.6)$$

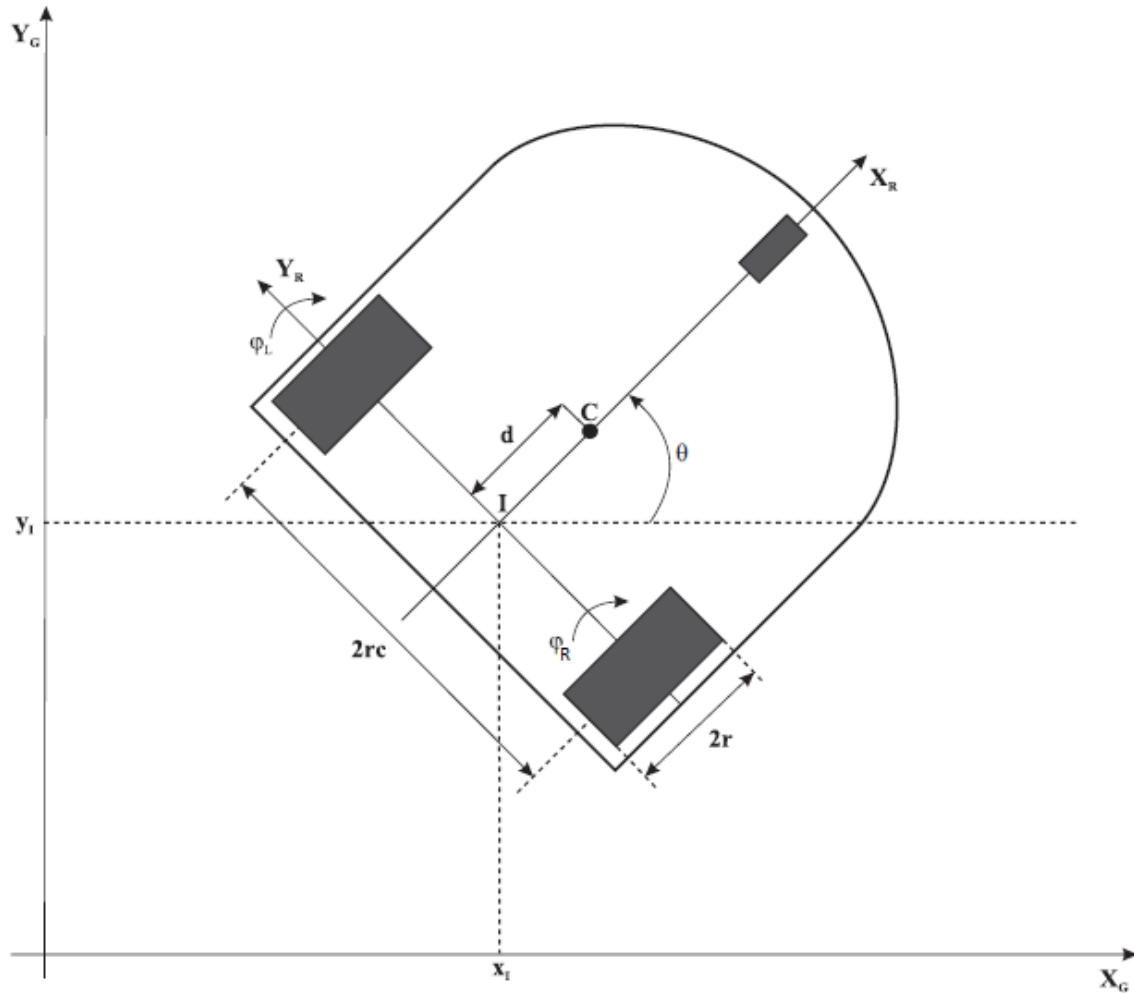
U daljem tekstu brzinu i položaj robota u globalnom koordinatnom sistemu označavaćemo bez

subskripta G , $\dot{\mathbf{q}} = \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix}$ i $\mathbf{q} = \begin{bmatrix} x \\ y \\ \theta \end{bmatrix}$.

O diferencijalnom pogonu mobilnog robota direktne kinematike će biti reči u sledećem odeljku.

3.1.1 Model direktne kinematike

Pretpostavimo da imamo diferencijalni pogon mobilnog robota koji ima dva točka sa radijusom r postavljena na udaljenosti r_c od centra robota kao na Slici 3.2:



Slika 3.2: Model diferencijalnog pogona mobilnog robota

Dalje u ovoj tezi koristise sledeće oznake:

I : Presek ose simetrije sa osom pogonskih točkova

C : Centar mase platforme

d : Rastojanje između centra mase i ose točka u pravcu x -ose

r_c : Rastojanje točka od ose simetrije robota u pravcu y -ose (lateralna osa)

r : Radijus točkova

$\dot{\phi}_R$: Brzina rotacije desnog točka

$\dot{\phi}_L$: Brzina rotacije levog točka

v : Translatorna brzina platforme u lokalnom sistemu

ω : Brzina rotacije platforme u lokalnom i globalnom sistemu

Problem direktne kinematike može se opisati kao problem pronalaženja sledeće funkcije:

$$\dot{\mathbf{q}} = \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix} = f(\dot{\phi}_R, \dot{\phi}_L, r_c, r, \theta) \quad (3.7)$$

Brzina točkova u koordinatnom sistemu robota je $r\dot{\phi}$, prema tome translatorska brzina robota jednaka je srednjoj brzini:

$$v = v_R = r \frac{\dot{\phi}_R + \dot{\phi}_L}{2} \quad (3.8)$$

A ugaona brzina je jednaka:

$$\omega = \omega_R = \dot{\theta} = \frac{r}{2r_c} (\dot{\phi}_R - \dot{\phi}_L) \quad (3.9)$$

Kako je $\dot{\mathbf{q}}_G = \mathbf{R}(\theta)^{-1} \dot{\mathbf{q}}_R$, celokupan model, što je u stvari brzina robota u globalnom sistemu, jednak je:

$$\dot{\mathbf{q}}_G = \mathbf{R}(\theta)^{-1} \frac{r}{2} \begin{bmatrix} \dot{\phi}_R + \dot{\phi}_L \\ 0 \\ \dot{\phi}_R - \dot{\phi}_L \\ r_c \end{bmatrix} \quad (3.10)$$

Inverzna rotaciona matrica jednaka je:

$$\mathbf{R}(\theta)^{-1} = \begin{bmatrix} \cos(\theta) & -\sin(\theta) & 0 \\ \sin(\theta) & \cos(\theta) & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (3.11)$$

Oдавде vidimo da je brzina robota u globalnom sistemu jednaka:

$$\dot{\mathbf{q}} = \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} \cos(\theta) & -\sin(\theta) & 0 \\ \sin(\theta) & \cos(\theta) & 0 \\ 0 & 0 & 1 \end{bmatrix} \frac{r}{2} \begin{bmatrix} \dot{\phi}_R + \dot{\phi}_L \\ 0 \\ \dot{\phi}_R - \dot{\phi}_L \\ r_c \end{bmatrix} = \begin{bmatrix} r \frac{\dot{\phi}_R + \dot{\phi}_L}{2} \cos(\theta) \\ r \frac{\dot{\phi}_R + \dot{\phi}_L}{2} \sin(\theta) \\ \frac{r}{2r_c} (\dot{\phi}_R - \dot{\phi}_L) \end{bmatrix} \quad (3.12)$$

Jednačina iznad je opšta jednačina direktne kinematike diferencijalnog pogonskog mobilnog robota. Još jedna alternativna forma kinematskog modela može se dobiti predstavljanjem brzina robota u globalnom sistemu u zavisnosti od linearne i ugaone brzine u robotskom koordinatnom sistemu.

$$\dot{\mathbf{q}} = \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} \cos\theta & 0 \\ \sin\theta & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} v \\ \omega \end{bmatrix} \quad (3.13)$$

Budući da je translatorna brzina u robotskom koordinatnom sistemu jednaka $v = r \frac{\omega_R + \omega_L}{2}$, a ugaona brzina $\omega = \frac{r}{2r_c} (\omega_R - \omega_L)$, njihova veza se može predstaviti u matričnoj formi:

$$\begin{bmatrix} v \\ \omega \end{bmatrix} = \begin{bmatrix} \frac{r}{2} & \frac{r}{2} \\ r & -r \\ \frac{2r_c}{2r_c} & \frac{2r_c}{2r_c} \end{bmatrix} \begin{bmatrix} \omega_R \\ \omega_L \end{bmatrix} \quad (3.14)$$

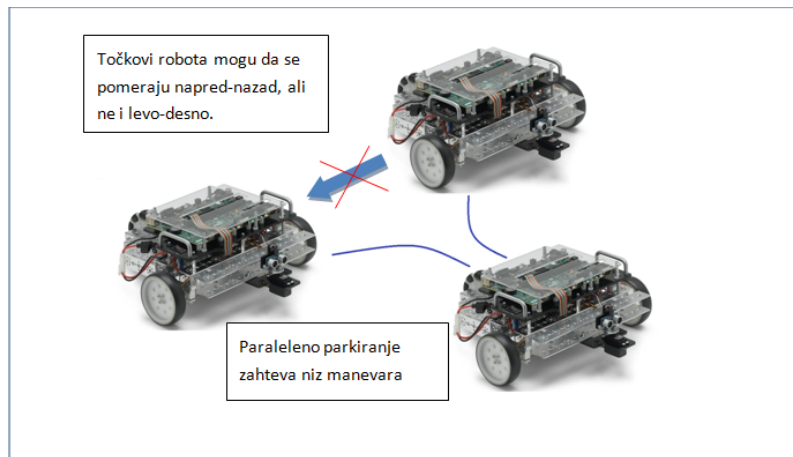
$$V = \begin{bmatrix} \frac{r}{2} & \frac{r}{2} \\ r & -r \\ \frac{2r_c}{2r_c} & \frac{2r_c}{2r_c} \end{bmatrix} \quad (3.15)$$

3.1.2. Neholonomska ograničenje kretanja robota

Sledeće pretpostavke vezane za kretanje točkova razmatranog mobilnog robota predstavljaju kinematska ograničenja robota [28]:

- Kretanje po horizontalnoj površini
- Tačka dodira između točkova i zemlje
- Točkovi su nepromenljivi
- Čisto kotrljanje što znači da postoji trenutni centar nulte brzine prilikom tačke dodira između točkova i zemlje
- Nema proklizavanja
- Nema trenja prilikom rotacije oko tačke dodira
- Ose upravljanja su ortogonalne u odnosu na površinu
- Točkovi su povezani preko šasije

Imajući u vidu gore navedene pretpostavke o kretanju točkova, robot ima posebno ograničenje koje se zove *neholonomska* ograničenje. Neholonomska ograničenje je ograničenje na tela koja imaju brzinu. U slučaju diferencijalnog pogona mobilnog robota, ovo znači da točkovi robota mogu da se kreću samo u nekim pravcima (napred i nazad), ali ne u svim (levo ili desno) kao što je prikazano na Slici 3.3:



Slika 3.3: Neholonomska ograničenje kretanja robota

Postojanje ovakvih ograničenja u sistemu izazvaće pojedine poteškoće u planiranju kretanja i kontroli ovakvih sistema o čemu će biti reči u jednom od narednih odeljaka u ovom radu. Jednačina neholonomskog ograničenja diferencijalnog pogona mobilnog robota prikazanog na Slici 3.1 dobija se iz sledećih pretpostavki [28][29]:

- Ograničenje gde nema bočnog klizanja:

Ovo znači da se točkovi robota mogu kretati samo napred nazad ali ne mogu se kretati levo i desno. U robotskom koordinatnom sistemu, ovaj uslov znači da je brzina centra ose simetrije robota I u pravcu (lateralne) bočne ose Y_R jednaka nuli

$$\dot{y}_I^R = 0 \quad (3.16)$$

Koristeći ortogonalnu rotacionu matricu $R(\theta)$, brzina centra robota u pravcu y_r ose u robotskom koordinatnom sistemu biće:

$$\dot{y}_I^R = -\dot{x} \sin \theta + \dot{y} \cos \theta \quad (3.17)$$

Odavde dobijamo jednačinu ograničenja bočnog klizanja:

$$-\dot{x} \sin \theta + \dot{y} \cos \theta = 0 \quad (3.18)$$

Ovo ograničenje znači da brzina centralne tačke robota ima pravac ose simetrije kao i da je kretanje ortogonalne ravni jednako nuli.

- Ograničenje čistog kotrljanja:

Ovo ograničenje prikazuje točkove koji ne proklizavaju.

Koristeći ortogonalnu rotacionu matricu $\mathbf{R}(\theta)$, brzina centra robota u pravcu X_R u robotskom koordinatnom sistemu biće:

$$\dot{x}_R = \dot{x}\cos\theta + \dot{y}\sin\theta \quad (3.19)$$

Budući da je \dot{x}_R jednako translatoronoj srednjoj brzini, jer nema bočnog klizanja, $\dot{x}_R = r \frac{\dot{\phi}_R + \dot{\phi}_L}{2}$, brzinu robota u robotskom koordinatnom sistemu možemo napisati kao :

$$\mathbf{v}_R = \begin{bmatrix} r \frac{\dot{\phi}_R + \dot{\phi}_L}{2} \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} \dot{x}\cos\theta + \dot{y}\sin\theta \\ -\dot{x}\sin\theta + \dot{y}\cos\theta \\ 0 \end{bmatrix} \quad (3.20)$$

Oдавде dobijamo jednačine ograničenja čistog kotrljanja :

$$\dot{x}\cos\theta + \dot{y}\sin\theta + r_c\dot{\theta} = r\dot{\phi}_R \quad (3.21)$$

$$\dot{x}\cos\theta + \dot{y}\sin\theta - r_c\dot{\theta} = r\dot{\phi}_L \quad (3.22)$$

Ova tri jednačine ograničenja (3.19), (3.21) (3.22) se mogu predstaviti u matričnoj formi na sledeći način:

$$\begin{bmatrix} \dot{\phi}_R \\ \dot{\phi}_L \end{bmatrix} = \begin{bmatrix} \cos\theta & \sin\theta & r_c \\ \cos\theta & \sin\theta & -r_c \end{bmatrix} \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix} \quad (3.23)$$

$$\begin{bmatrix} -\sin\theta & \cos\theta & 0 \end{bmatrix} \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix} = \mathbf{A}(q)\dot{\mathbf{q}} = 0 \quad (3.24)$$

Ovakav izraz za neholonomsko ograničenje je koristan kada hoćemo da uzmemo u obzir ograničenja kod dinamičkog modeliranja.

3.2. DINAMIČKO MODELIRANJE MOBILNOG ROBOTA SA DIFERENCIJALNIM POGONOM

Za dinamičko modeliranje robota s diferencijalnim pogonom, najčešće se koristi Langražeov metod [31-36] ili pak Nutonov-Ojlerov pristup [37-39]. Takođe, pojedini istraživači preporučuju Kanov metod kao varijabilni pristup modelovanja mobilnog robota sa diferencijalnim pogonom [40].

Kod Njutnovog metoda, uzimaju se u obzir i sile koje deluju na sistem i sile ograničenja. Sile koje deluju na sistem su pogoneske sile aktuatora, dok sile ograničenja predstavljaju sile trenja između točkova robota i podloge. Međutim, u sistemu sa povezanim elementima, komponente mogu međusobno delovati jedne na druge preko zupčanika, opruga i frikcionih elementa [39]. Prema tome, moramo da uzmemo u obzir sve ove sile. Jasno je da se ovim metodom da imamo poteškoće, jer često ove sile nisu lako merljive.

Langražeov pristup je snažan i svestran metod za formulisanje jednačina kretanja za holonomske sisteme. Međutim, kod neholonomskih sistema, uobičajno se uvode jednačne ograničenja u dinamičke jednačine pomoću Langražeovih multiplikatora. Ovi multiplikatori nisu konstantni i uglavnom su u funkciji svih generalizovanih koordinata kao i vremena. Oni predstavljaju skup nepoznatih čije vrednosti mogu biti dobijene kao deo rešenja. U cilju smanjenja kompleksnosti izračunavanja, dodatni metodi su predloženi da uklone prisustvo ovih multiplikatora iz dinamičkih jednačina datog sistema [41].

Definisanje jednačine kretanja, odnosno dinamike robota, ima suštinski značaj u analizi, dizajnu i kontroli robota. U globalu dinamičko modeliranje se može definisati kao proučavanje kretanja sistema gde se modeliraju sile, a mogu se uključiti i energija kao i brzine koje su u vezi sa kretanjem. Osnovna razlika između dinamičkog i kinematskog modeliranja jeste u tome da kod kinematskog modeliranja proučavamo kretanje ne uzimajući u obzir sile koje utiču na kretanje, odnosno bavimo se samo geometrijskim vezama koje upravljaju sistemom.

Pretpostavimo da sistem mobilnih robota ima n -dimenzionalnu konfiguraciju prostora D sa koordinatama (q_1, q_2, \dots, q_n) podložan m ograničenjima može biti predstavljen pomoću sledeće uopštene jednačine dinamike:

$$\mathbf{M}(q)\ddot{\mathbf{q}} + \mathbf{C}(q, \dot{\mathbf{q}})\dot{\mathbf{q}} + \mathbf{F}(\dot{\mathbf{q}}) + \mathbf{G}(q) + \boldsymbol{\tau}_d = \mathbf{B}(q)\boldsymbol{\tau} - \mathbf{A}^T(q)\boldsymbol{\lambda} \quad (3.25)$$

gde su:

$\mathbf{M}(q)$ - simetrično pozitivna definisana matrica

$\mathbf{C}(q, \dot{\mathbf{q}})$ - centripetalna i Koriolisova matrica

$\mathbf{F}(\dot{\mathbf{q}})$ - matrica površinskog trenja

$\mathbf{G}(q)$ - vektor gravitacije

τ_d - označene, ograničene nepoznate smetnje uključujući i nestrukturiranu, nemodeliranu dinamiku

$B(q)$ - ulazna transformaciona matrica

τ - ulazni vektor

$A^T(q)$ - matrica povezana sa ograničenjima

λ - vektor ograničenja snage

U narednom odeljku biće predstavljen Lagranžov metod za izražavanje dinamike ovog robota. Ovaj metod je pogodan za formulisanje i implementiranje

3.2.1. Lagranžev metod dinamike

Analitička dinamika je metod u dinamici koji tretira sistem kao celinu koja se bavi skalarnim veličinama kao što su kinetička i potencijalna energija sistema. Lagranže je predložio metod koji pruža snažan i prilagodljiv način za formulisanje jednačina kretanja za svaki dinamički sistem. Lagranžeove jednačine su diferencijalne jednačine u kojima se uzima u obzir energija sistema kao i rad koji se izvršava u datom trenutku vremena. Izvedena jednačina iz Lagranžeove jednačine za holonomske sisteme zahteva da generalizovane koordinate budu nezavisne. Međutim, za neholonomske sisteme mora postojati veći broj generalizovanih koordinata od broja stepena slobode zbog ograničenja kretanja sistema [30].

Lagranžeova jednačina može se napisati u sledećoj formi:

$$\frac{d}{dt} \left(\frac{\partial \mathcal{L}}{\partial \dot{q}_i} \right) - \frac{\partial \mathcal{L}}{\partial q_i} = F_i - A^T(q) \lambda_j, i = 1, 2, \dots, n \quad (3.26)$$

gde su:

q_1, q_2, \dots, q_n generalizovane koordinate.

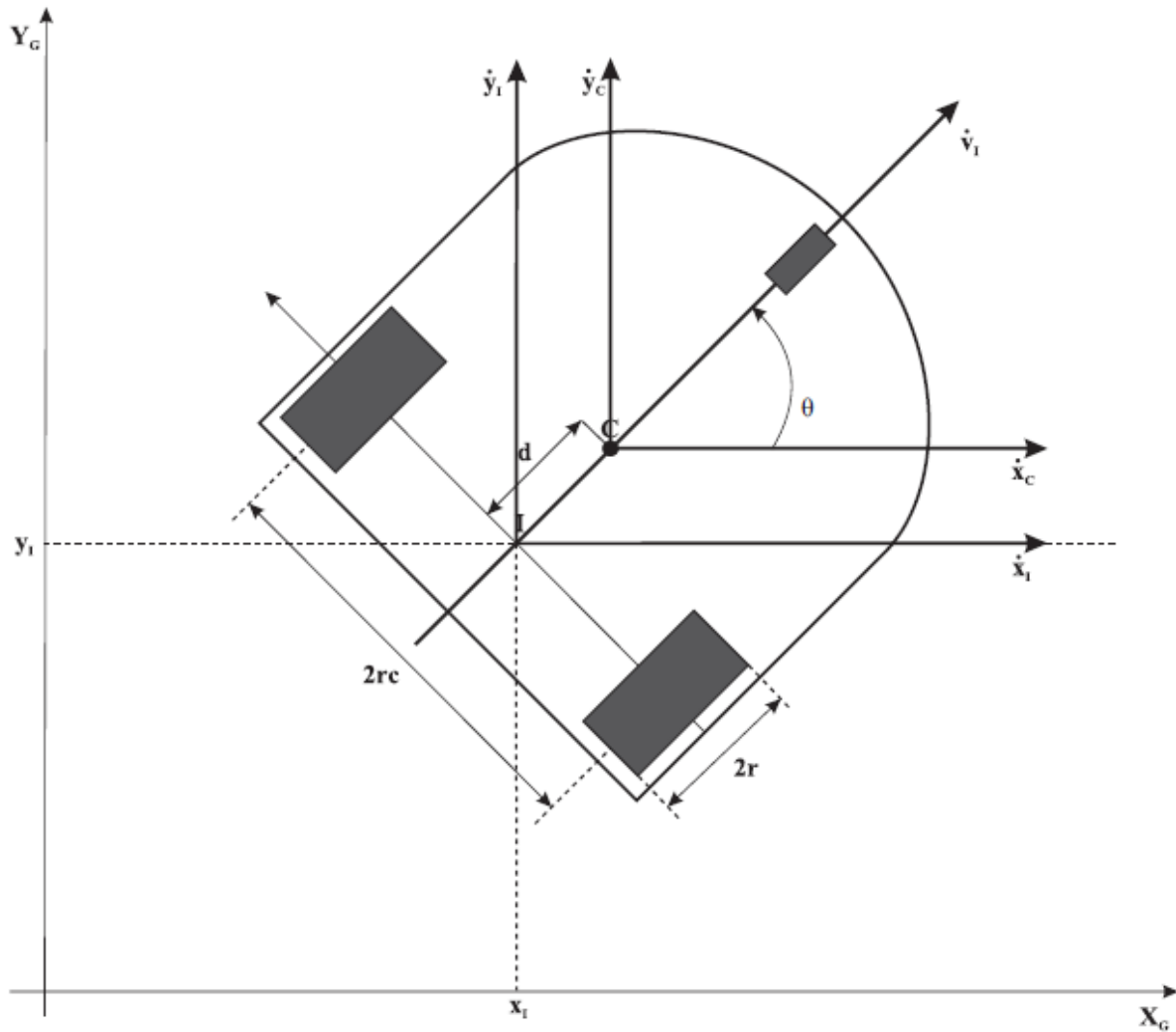
$\mathcal{L} = E_k - E_p$ je Lagranže što predstavlja razliku kinetičke i potencijalne energije sistema.

λ_j je Lagranžev umnožilac koji povezuje ograničenja sa silom ograničenja.

A^T je matrica ograničenja

F_i su generalizovane sile sistema.

Prvi korak u pronalaženju jednačine dinamike je da se sračunaju kinetička i potencijalna energija sistema. Potencijalna energija sistema jednaka je nuli zato što je kretanje ograničeno samo na zemlju. Funkcija kinetičke energije robota može biti izvedena iz brzine prikazanoj na Slici 3.4:



Slika 3.4: Brzina centra robota

Kako bismo našli brzine u tačkama I i C koristimo sledeće jednačine:

$$x_C = x + d \cos \theta \quad (3.27)$$

$$y_C = y + d \sin \theta \quad (3.28)$$

Odatle sledi:

$$\dot{x}_C = \dot{x} - d \dot{\theta} \sin \theta \quad (3.29)$$

$$\dot{y}_C = \dot{y} + d \dot{\theta} \cos \theta \quad (3.30)$$

Brzina centra mase robota C je:

$$v_C = \dot{x} \vec{i} + \dot{y} \vec{j} - d \dot{\theta} \sin \theta \vec{i} + d \dot{\theta} \cos \theta \vec{j} \quad (3.31)$$

$$v_C = (\dot{x} - d \dot{\theta} \sin \theta) \vec{i} + (\dot{y} + d \dot{\theta} \cos \theta) \vec{j} \quad (3.32)$$

Pri složenom kretanju krutog tela ukupna kinetička energija je suma kinetičkih energija translatornog kretanja centra mase i rotacionog kretanja tela.

Kinetička energija robotske platforme je:

$$E_k = \frac{1}{2}mv_c^2 + \frac{1}{2}I_C\dot{\theta}^2 \quad (3.33)$$

$$\begin{aligned} E_k &= \frac{1}{2}m \left[(\dot{x} - d\dot{\theta}\sin\theta)^2 + (\dot{y} + d\dot{\theta}\cos\theta)^2 \right] + \frac{1}{2}I_C\dot{\theta}^2 \\ &= \frac{1}{2}m \left[\dot{x}^2 - 2\dot{x}d\dot{\theta}\sin\theta + d^2\dot{\theta}^2\sin^2\theta + \dot{y}^2 + 2\dot{y}d\dot{\theta}\cos\theta \right. \\ &\quad \left. + d^2\dot{\theta}^2\cos^2\theta \right] + \frac{1}{2}I_C\dot{\theta}^2 \end{aligned} \quad (3.34)$$

$$E_k = \frac{1}{2}m\dot{x}^2 + \frac{1}{2}m\dot{y}^2 - m\dot{x}d\dot{\theta}\sin\theta + m\dot{y}d\dot{\theta}\cos\theta + \frac{1}{2}md^2\dot{\theta}^2 + \frac{1}{2}I_C\dot{\theta}^2 \quad (3.35)$$

Iz teoreme paralelne ose (Štajnerove teoreme) možemo da izvedemo sledeće:

$$I_C + md^2 = I_I = I \quad (3.36)$$

Prema tome kinetička energija robotske platforme će biti:

$$E_k = \frac{1}{2}m\dot{x}^2 + \frac{1}{2}m\dot{y}^2 + \frac{1}{2}I\dot{\theta}^2 + m\dot{x}d\dot{\theta}\sin\theta + m\dot{y}d\dot{\theta}\cos\theta \quad (3.37)$$

Iz jednačine kinetičke energije prikazane iznad, znajući da je potencijalna energija jednaka nuli, Lagranže postaje:

$$\mathcal{L} = \frac{1}{2}m\dot{x}^2 + \frac{1}{2}m\dot{y}^2 + \frac{1}{2}I\dot{\theta}^2 - m\dot{x}d\dot{\theta}\sin\theta + m\dot{y}d\dot{\theta}\cos\theta \quad (3.38)$$

Generalizovane koordinate koje se koriste u Lagranžeovoj formuli su predstavljene na sledeći način:

$$\mathbf{q} = [x, y, \theta]^T \quad (3.39)$$

$$\dot{\mathbf{q}} = [\dot{x}, \dot{y}, \dot{\theta}]^T \quad (3.40)$$

Korak po korak pristup za pronalaženje jednačina dinamike koristeći gore predstavljene koordinate kao i Lagranže može se predstaviti kao [30][39]:

$$\frac{\partial \mathcal{L}}{\partial \dot{x}} = m\dot{x} - md\dot{\theta}\sin\theta \quad (3.41)$$

$$\frac{\partial \mathcal{L}}{\partial \dot{y}} = m\dot{y} + md\dot{\theta}\cos\theta \quad (3.42)$$

$$\frac{\partial \mathcal{L}}{\partial \dot{\theta}} = I\dot{\theta} - m\dot{x}d\sin\theta + m\dot{y}d\cos\theta \quad (3.43)$$

Prvi izvod po vremenu jednačina iznad jednak je:

$$\frac{d}{dt}\left(\frac{\partial \mathcal{L}}{\partial \dot{x}}\right) = m\ddot{x} - md\ddot{\theta}\sin\theta - md\dot{\theta}^2\cos\theta \quad (3.44)$$

$$\frac{d}{dt}\left(\frac{\partial \mathcal{L}}{\partial \dot{y}}\right) = m\ddot{y} + md\ddot{\theta}\cos\theta - md\dot{\theta}^2\sin\theta \quad (3.45)$$

$$\frac{d}{dt}\left(\frac{\partial \mathcal{L}}{\partial \dot{\theta}}\right) = I\ddot{\theta} - md\ddot{x}\sin\theta - md\dot{x}\dot{\theta}\cos\theta + md\dot{y}\cos\theta - m\dot{y}d\dot{\theta}\sin\theta \quad (3.46)$$

Ostala izvođenja potrebna za Lagranžeove jednačine su:

$$\frac{\partial \mathcal{L}}{\partial x} = 0 \quad (3.47)$$

$$\frac{\partial \mathcal{L}}{\partial y} = 0 \quad (3.48)$$

$$\frac{\partial \mathcal{L}}{\partial \theta} = -m\dot{x}d\dot{\theta}\cos\theta - m\dot{y}d\dot{\theta}\sin\theta \quad (3.49)$$

Ako se jednačine iznad zamene u u Lagranžeovu jednačinu dobija se:

$$m\ddot{x} - md\ddot{\theta}\sin\theta - md\dot{\theta}^2\cos\theta = F_x + C_x \quad (3.50)$$

$$m\ddot{y} + md\ddot{\theta}\cos\theta - md\dot{\theta}^2\sin\theta = F_y + C_y \quad (3.51)$$

$$I\ddot{\theta} - md\ddot{x}\sin\theta - md\dot{x}\dot{\theta}\cos\theta + md\dot{y}\cos\theta - m\dot{y}d\dot{\theta}\sin\theta + m\dot{x}d\dot{\theta}\cos\theta + m\dot{y}d\dot{\theta}\sin\theta = M + C_\theta \quad (3.52)$$

Nakon uprošćavanja jednačina iznad dobijamo:

$$m\ddot{x} - md\ddot{\theta}\sin\theta - md\dot{\theta}^2\cos\theta = F_x + C_x \quad (3.53)$$

$$m\ddot{y} + md\ddot{\theta}\cos\theta - md\dot{\theta}^2\sin\theta = F_y + C_y \quad (3.54)$$

$$I\ddot{\theta} - md\ddot{x}\sin\theta + md\dot{y}\cos\theta = M + C_\theta \quad (3.55)$$

gde je:

F_x pogonska sila u pravcu x -ose.

F_y pogonska sila u pravcu y -ose.

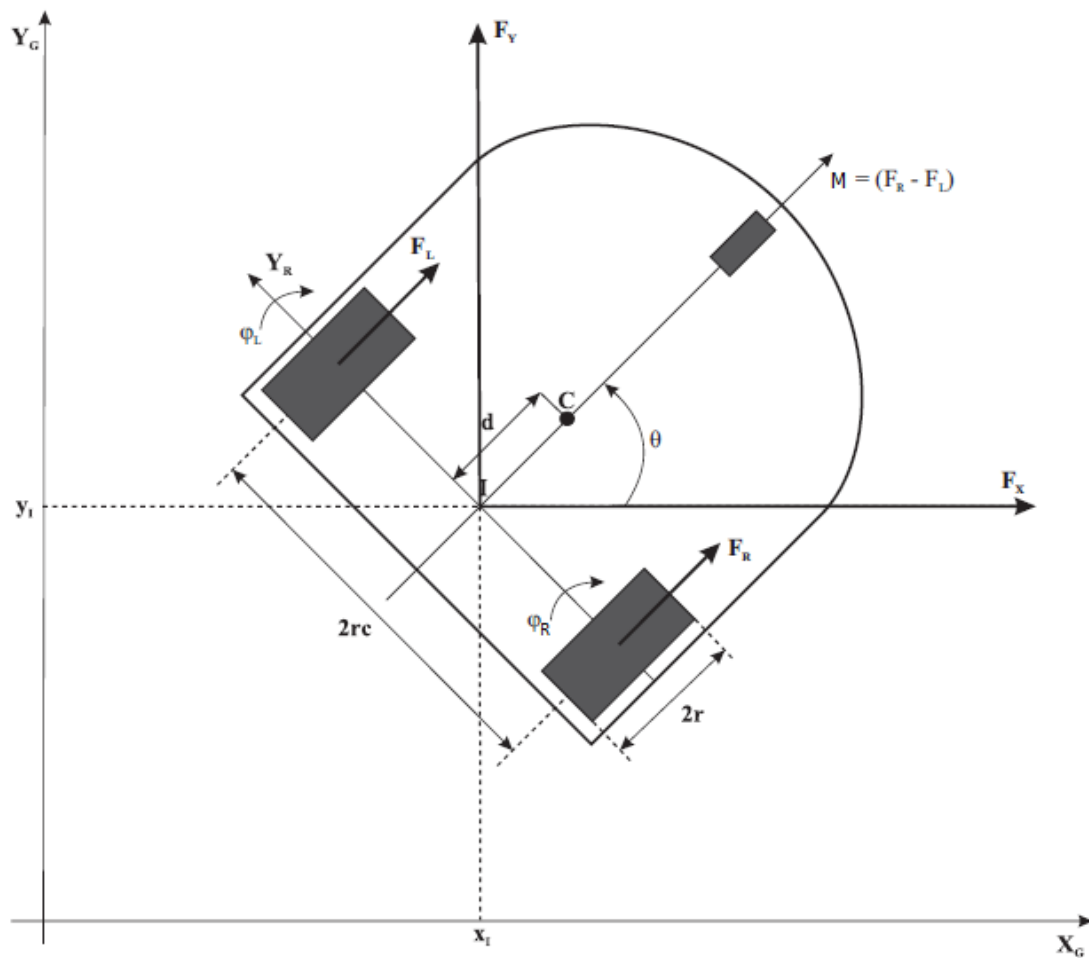
M pogonski obrtni moment sile robota

C_X, C_Y, C_θ su sile ograničenja u pravcu x, y i θ ose.

Ako se jednačine iznad predstave pomoću matrice dobija se:

$$\begin{bmatrix} m & 0 & -md\sin\theta \\ 0 & m & md\cos\theta \\ -md\sin\theta & md\cos\theta & I_C + md^2 \end{bmatrix} \ddot{q} + \begin{bmatrix} 0 & 0 & -md\dot{\theta}\cos\theta \\ 0 & 0 & -md\dot{\theta}\sin\theta \\ 0 & 0 & 0 \end{bmatrix} \dot{q} = \begin{bmatrix} F_x \\ F_y \\ M \end{bmatrix} + \begin{bmatrix} C_X \\ C_Y \\ C_\theta \end{bmatrix} \quad (3.56)$$

Sile u x, y i θ pravcu mogu da se povežu sa pogonskim obrtnim momentom sile točkova na osnovu dijagrama sila prikazanog na Slici 3.5:



Slika 3.5: Dijagram sila

$$F_R = \frac{\tau_R}{r} \quad (3.57)$$

$$F_L = \frac{\tau_L}{r} \quad (3.58)$$

$$F_X = \left(\frac{\tau_R}{r} + \frac{\tau_L}{r} \right) \cos\theta = \frac{\cos\theta}{r} (\tau_R + \tau_L) \quad (3.59)$$

$$F_y = \left(\frac{\tau_R}{r} + \frac{\tau_L}{r} \right) \sin\theta = \frac{\sin\theta}{r} (\tau_R + \tau_L) \quad (3.60)$$

$$M = (F_R - F_L)r_c = \frac{r_c}{r} (\tau_R - \tau_L) \quad (3.61)$$

Na osnovu jednačina iznad, može se napisati matrica ulazne sile u jednačini dinamike sistema na sledeći način:

$$\begin{bmatrix} F_x \\ F_y \\ M \end{bmatrix} = \begin{bmatrix} \frac{\cos\theta}{r} (\tau_R + \tau_L) \\ \frac{\sin\theta}{r} (\tau_R + \tau_L) \\ \frac{r_c}{r} (\tau_R - \tau_L) \end{bmatrix} = \frac{1}{r} \begin{bmatrix} \cos\theta & \cos\theta \\ \sin\theta & \sin\theta \\ r_c & -r_c \end{bmatrix} \begin{bmatrix} \tau_R \\ \tau_L \end{bmatrix} \quad (3.62)$$

$$\mathbf{B}(q)\boldsymbol{\tau} = \frac{1}{r} \begin{bmatrix} \cos\theta & \cos\theta \\ \sin\theta & \sin\theta \\ r_c & -r_c \end{bmatrix} \begin{bmatrix} \tau_R \\ \tau_L \end{bmatrix}$$

Na osnovu izvedenih jednačina iznad, konačan oblik jednačine diferencijalnog pogona mobilnog robota izgleda:

$$\begin{bmatrix} m & 0 & -md\sin\theta \\ 0 & m & md\cos\theta \\ -md\sin\theta & md\cos\theta & I_c + 2md^2 \end{bmatrix} \ddot{\mathbf{q}} + \begin{bmatrix} 0 & 0 & -md\dot{\theta}\cos\theta \\ 0 & 0 & -md\dot{\theta}\sin\theta \\ 0 & 0 & 0 \end{bmatrix} \dot{\mathbf{q}} \\ = \frac{1}{r} \begin{bmatrix} \cos\theta & \cos\theta \\ \sin\theta & \sin\theta \\ r_c & -r_c \end{bmatrix} \begin{bmatrix} \tau_R \\ \tau_L \end{bmatrix} - \mathbf{A}^T(q)\boldsymbol{\lambda}_j \quad (3.63)$$

Ako se uporedi jednačina iznad sa opštom jednačinom dinamike robota (3.25), dobijaju se sledeći parametri sistema diferencijalnog pogona mobilnog robota:

$$\mathbf{M}(q) = \begin{bmatrix} m & 0 & -md\sin\theta \\ 0 & m & md\cos\theta \\ -md\sin\theta & md\cos\theta & I \end{bmatrix}$$

$$\mathbf{C} = \begin{bmatrix} 0 & 0 & -md\dot{\theta}\cos\theta \\ 0 & 0 & -md\dot{\theta}\sin\theta \\ 0 & 0 & 0 \end{bmatrix}$$

$\mathbf{F}(\dot{\mathbf{q}}) = 0$ uzima se da je nula u ovoj jednačini

$\mathbf{G}(q) = 0$ kretanje je ograničeno na zemlju

$\boldsymbol{\tau}_d = 0$ uzima se da je nula u ovoj jednačini

$$\boldsymbol{\tau} = \begin{bmatrix} \tau_R \\ \tau_L \end{bmatrix}$$

$$\mathbf{A}^T(q) = \begin{bmatrix} -\sin\theta \\ \cos\theta \\ 0 \end{bmatrix}$$

$$\mathbf{B}(q) = \frac{1}{r} \begin{bmatrix} \cos\theta & \cos\theta \\ \sin\theta & \sin\theta \\ r_c & -r_c \end{bmatrix}$$

3.2.1.1. Redukovan dinamički model

Navedeni sistem može da se transformiše tako da više odgovara za svrhe kontrole i simulacije, eliminacijom parametara ograničavanja iz jednačine. Ovo se postiže pomoću dve matrice prikazane ispod:

$$\mathbf{u}(t) = \begin{bmatrix} v \\ \omega \end{bmatrix} = \begin{bmatrix} u_1 \\ u_2 \end{bmatrix} \quad (3.64)$$

$$\mathbf{S}(q) = \begin{bmatrix} \cos\theta & 0 \\ \sin\theta & 0 \\ 0 & 1 \end{bmatrix} \quad (3.65)$$

Iz jednačine direktne kinematike vidi se da je $\mathbf{S}(q)$ matrica modifikovana matrica direktne kinematike koja ima dva parametra za brzinu. Odavde sledi sledeća jednačina sistema:

$$\dot{\mathbf{q}} = \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix} = \mathbf{S}(q)\mathbf{u}(t) = \begin{bmatrix} \cos\theta & 0 \\ \sin\theta & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} v \\ \omega \end{bmatrix} \quad (3.66)$$

Vrlo jednostavno se može dokazati da $\mathbf{S}(q)$ matrica ima sledeću vezu sa $\mathbf{A}(q)$ matricom:

$$\mathbf{S}^T(q)\mathbf{A}^T(q) = \mathbf{0} \quad (3.67)$$

Jednačina iznad je korisna za eliminaciju parametre ograničenja iz glavne jednačine dinamike kao što će se videti u narednom koraku. Diferenciranjem jednačine (3.67) dobijamo:

$$\ddot{\mathbf{q}} = \dot{\mathbf{S}}(q)\mathbf{u}(t) + \mathbf{S}(q)\dot{\mathbf{u}}(t) \quad (3.68)$$

Zamenom jednačine iznad u (3.25) dobićemo sledeću jednačinu:

$$\begin{aligned} \mathbf{M}(q)[\dot{\mathbf{S}}(q)\mathbf{u}(t) + \mathbf{S}(q)\dot{\mathbf{u}}(t)] + \mathbf{C}(q, \dot{q})[\mathbf{S}(q)\mathbf{u}(t)] + \mathbf{F}(\dot{q}) + \mathbf{G}(q) + \boldsymbol{\tau}_d \\ = \mathbf{B}(q)\boldsymbol{\tau} - \mathbf{A}^T(q)\boldsymbol{\lambda} \end{aligned} \quad (3.69)$$

$$\begin{aligned} \mathbf{M}(q)\dot{\mathbf{S}}(q)\mathbf{u}(t) + \mathbf{M}(q)\mathbf{S}(q)\dot{\mathbf{u}}(t) + \mathbf{C}(q, \dot{q})\mathbf{S}(q)\mathbf{u}(t) + \mathbf{F}(\dot{q}) + \mathbf{G}(q) + \boldsymbol{\tau}_d \\ = \mathbf{B}(q)\boldsymbol{\tau} - \mathbf{A}^T(q)\boldsymbol{\lambda} \end{aligned} \quad (3.70)$$

Sledeći korak u eliminisanju matrice ograničenja $\mathbf{A}^T(q)\boldsymbol{\lambda}$ je da se jednačina (3.70) pomnoži sa $\mathbf{S}^T(q)$ na sledeći način:

$$\begin{aligned}
& [\mathbf{S}^T(q)\mathbf{M}(q)\mathbf{S}(q)]\dot{\mathbf{u}}(t) + [\mathbf{S}^T(q)\mathbf{M}(q)\dot{\mathbf{S}}(q) + \mathbf{S}^T(q)\mathbf{C}(q, \dot{q})\mathbf{S}(q)]\mathbf{u}(t) \\
& + \mathbf{S}^T(q)\mathbf{F}(\dot{q}) + \mathbf{S}^T(q)\mathbf{G}(q) + \mathbf{S}^T(q)\boldsymbol{\tau}_d \\
& = \mathbf{S}^T(q)\mathbf{B}(q)\boldsymbol{\tau} - \mathbf{S}^T(q)\mathbf{A}^T(q)\boldsymbol{\lambda}
\end{aligned} \tag{3.71}$$

Kao što može da se vidi u jednačini iznad, $\mathbf{S}^T(q)\mathbf{A}^T(q)$ je jednako nuli na osnovu jednačine (3.67). Tako smo eliminisali parametre ograničenja i jednačina dinamike sada izgleda ovako:

$$\begin{aligned}
& [\mathbf{S}^T(q)\mathbf{M}(q)\mathbf{S}(q)]\dot{\mathbf{u}}(t) + [\mathbf{S}^T(q)\mathbf{M}(q)\dot{\mathbf{S}}(q) + \mathbf{S}^T(q)\mathbf{C}(q, \dot{q})\mathbf{S}(q)]\mathbf{u}(t) \\
& + \mathbf{S}^T(q)\mathbf{F}(\dot{q}) + \mathbf{S}^T(q)\mathbf{G}(q) + \mathbf{S}^T(q)\boldsymbol{\tau}_d = \mathbf{S}^T(q)\mathbf{B}(q)\boldsymbol{\tau}
\end{aligned} \tag{3.72}$$

Koristeći odgovarajuće definicije, jednačinu iznad možemo predstaviti na sledeći način:

$$\bar{\mathbf{M}}(q)\dot{\mathbf{u}}(t) + \bar{\mathbf{C}}_m(q, \dot{q})\mathbf{u}(t) + \bar{\mathbf{F}}(\dot{q}) + \bar{\mathbf{G}}(q) + \bar{\boldsymbol{\tau}}_d = \bar{\mathbf{B}}(q)\boldsymbol{\tau} \tag{3.73}$$

$$\mathbf{S}^T(q)\mathbf{M}(q)\mathbf{S}(q) = \bar{\mathbf{M}}(q) \tag{3.74}$$

$$\mathbf{S}^T(q)\mathbf{M}(q)\dot{\mathbf{S}}(q) + \mathbf{S}^T(q)\mathbf{C}(q, \dot{q})\mathbf{S}(q) = \bar{\mathbf{C}}(q, \dot{q}) \tag{3.75}$$

$$\mathbf{S}^T(q)\mathbf{F}(\dot{q}) = \bar{\mathbf{F}}(\dot{q}) = 0 \tag{3.76}$$

$$\mathbf{S}^T(q)\mathbf{G}(q) = \bar{\mathbf{G}}(q) = 0 \tag{3.77}$$

$$\mathbf{S}^T(q)\boldsymbol{\tau}_d = \bar{\boldsymbol{\tau}}_d \tag{3.78}$$

$$\mathbf{S}^T(q)\mathbf{B}(q) = \bar{\mathbf{B}}(q) \tag{3.79}$$

$$\mathbf{S}^T(q)\mathbf{A}^T(q)\boldsymbol{\lambda} = 0 \tag{3.80}$$

Nova matrica mase $\bar{\mathbf{M}}(q)$ postaje:

$$\begin{aligned}
\bar{\mathbf{M}}(q) &= \mathbf{S}^T(q)\mathbf{M}(q)\mathbf{S}(q) \\
&= \begin{bmatrix} \cos\theta & \sin\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} m & 0 & -md\sin\theta \\ 0 & m & md\cos\theta \\ -md\sin\theta & md\cos\theta & I \end{bmatrix} \begin{bmatrix} \cos\theta & 0 \\ \sin\theta & 0 \\ 0 & 1 \end{bmatrix} \\
&= \begin{bmatrix} m\cos\theta & m\sin\theta & 0 \\ -md\sin\theta & md\cos\theta & I \end{bmatrix} \begin{bmatrix} \cos\theta & 0 \\ \sin\theta & 0 \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} m & 0 \\ 0 & I \end{bmatrix}
\end{aligned} \tag{3.81}$$

Nova matrica $\bar{\mathbf{C}}(q, \dot{q})$ postaje

$$\begin{aligned}
\bar{\mathbf{C}}(q, \dot{q}) &= \mathbf{S}^T(q) \mathbf{M}(q) \dot{\mathbf{S}}(q) + \mathbf{S}^T(q) \mathbf{C}_m^T(q, \dot{q}) \mathbf{S}(q) = \\
&= \begin{bmatrix} \cos\theta & \sin\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} m & 0 & -md\sin\theta \\ 0 & m & md\cos\theta \\ -md\sin\theta & md\cos\theta & I \end{bmatrix} \begin{bmatrix} -\dot{\theta}\sin\theta & 0 \\ \dot{\theta}\cos\theta & 0 \\ 0 & 0 \end{bmatrix} \\
&+ \begin{bmatrix} \cos\theta & \sin\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 0 & 0 & -md\dot{\theta}\cos\theta \\ 0 & 0 & -md\dot{\theta}\sin\theta \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \cos\theta & 0 \\ \sin\theta & 0 \\ 0 & 1 \end{bmatrix} \\
&= \begin{bmatrix} m\cos\theta & m\sin\theta & 0 \\ -md\sin\theta & md\cos\theta & I \end{bmatrix} \begin{bmatrix} -\dot{\theta}\sin\theta & 0 \\ \dot{\theta}\cos\theta & 0 \\ 0 & 0 \end{bmatrix} + \begin{bmatrix} 0 & 0 & -md\dot{\theta} \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \cos\theta & 0 \\ \sin\theta & 0 \\ 0 & 1 \end{bmatrix} \\
&= \begin{bmatrix} 0 & 0 \\ md\dot{\theta} & 0 \end{bmatrix} + \begin{bmatrix} 0 & -md\dot{\theta} \\ 0 & 0 \end{bmatrix} = \begin{bmatrix} 0 & -md\dot{\theta} \\ md\dot{\theta} & 0 \end{bmatrix}
\end{aligned} \tag{3.82}$$

Nova matrica $\bar{\mathbf{B}}(q)$ postaje:

$$\bar{\mathbf{B}}(q) = \mathbf{S}^T(q) \mathbf{B}(q) = \begin{bmatrix} \cos\theta & \sin\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \frac{1}{r} \begin{bmatrix} \cos\theta & \cos\theta \\ \sin\theta & \sin\theta \\ r_c & -r_c \end{bmatrix} = \frac{1}{r} \begin{bmatrix} 1 & 1 \\ r_c & -r_c \end{bmatrix} \tag{3.83}$$

Sada se jednačina 3.61 može napisati kao:

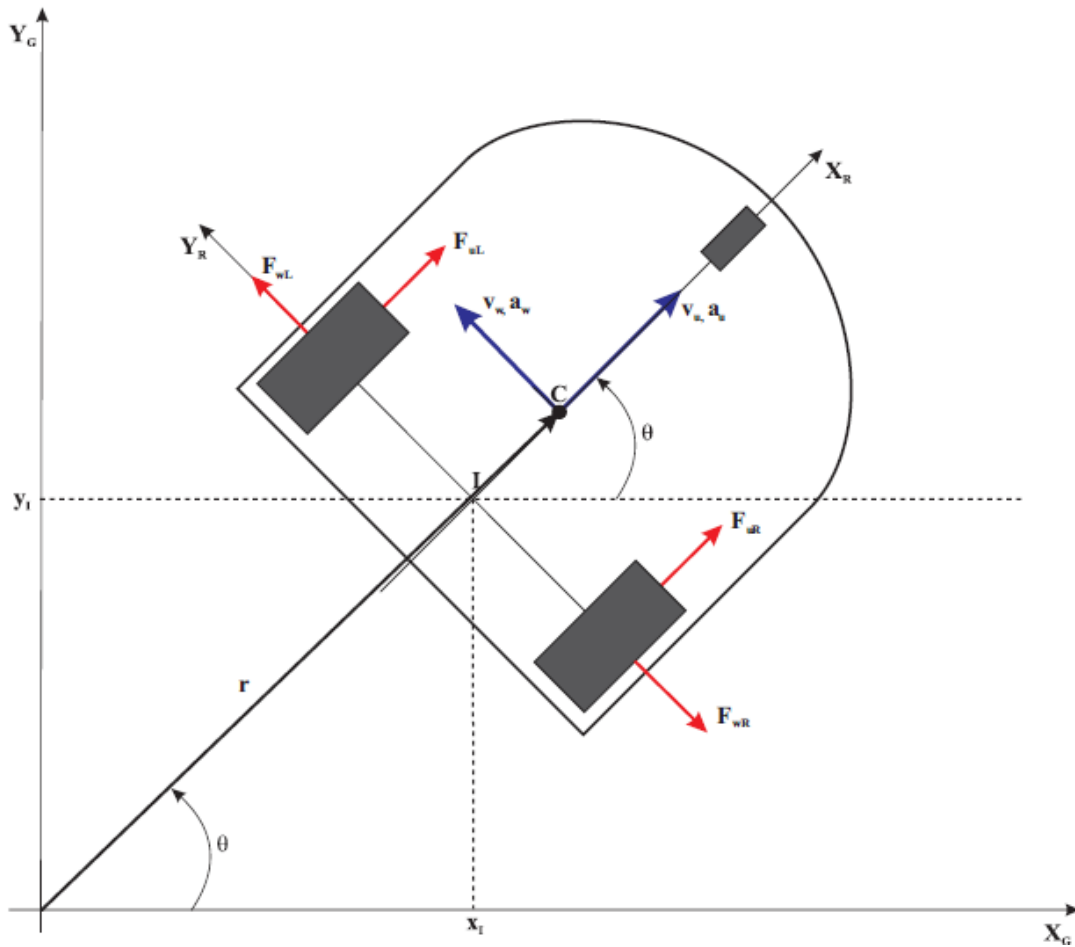
$$\begin{bmatrix} m & 0 \\ 0 & I \end{bmatrix} \begin{bmatrix} \dot{v} \\ \dot{\theta} \end{bmatrix} + \begin{bmatrix} 0 & -md\dot{\theta} \\ md\dot{\theta} & 0 \end{bmatrix} \begin{bmatrix} v \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} \frac{1}{r} & \frac{1}{r} \\ \frac{r_c}{r} & \frac{-r_c}{r} \end{bmatrix} \begin{bmatrix} \tau_R \\ \tau_L \end{bmatrix} \tag{3.84}$$

Pogonski momenti točkova su u korelaciji sa pogonskim obrtnim momentom i linearnom pogonskom silom robota, na sličan način kao i ugaona i linearna brzina robota sa ugaonim brzinama točkova.

$$\begin{bmatrix} F \\ M \end{bmatrix} = \begin{bmatrix} \frac{1}{r} & \frac{1}{r} \\ \frac{r_c}{r} & \frac{-r_c}{r} \end{bmatrix} \begin{bmatrix} \tau_R \\ \tau_L \end{bmatrix} \tag{3.85}$$

3.2.3. Njutn-Ojlerov metod dinamike

Prvi i jedan od najvažnijih koraka u Njutnovom dinamičkom modeliranju je crtanje dijagrama sila sistema i analiza sila koje deluju na sistem [30]. Dijagram sila diferencijalnog pogona mobilnog robota je prikazan na Slici 3.6:



Slika 3.6: Dijagram sile robota kod Njutnovog dinamičkog modeliranja

Sledeće oznake se nalaze na slici i koriste se kod Njutnovog dinamičkog modeliranja:

(v_u, v_w) : Predstavlja brzinu vozila u lokalnom sistemu. v_u je longitudinalna brzina, a v_w je lateralna brzina.

(a_u, a_w) : Predstavlja ubrzanje centra mase vozila.

(F_{u_l}, F_{u_r}) : Predstavlja longitudinalne sile koje deluju na vozilo od strane točkova.

(F_{w_l}, F_{w_r}) : Predstavlja bočne sile koje deluju na vozilo od strane točkova.

Kao što se može videti na dijagramu sile iznad, jedine sile koje deluju na robot su sile aktuatora koje deluju na točkove robota. Počinjemo izvođenje predstavljanjem pozicije robota pomoću polarnih koordinata. Ako pretpostavimo da je robot čvrsto telo, njegova pozicija može biti predstavljena pomoću ugla i radijusa:

$$\hat{r} = r e^{i\theta} \quad (3.86)$$

Diferenciranjem vektora pozicije prikazanog iznad dobijaju se brzina i ubrzanje robota:

$$\dot{\hat{r}} = \dot{r} e^{i\theta} + r \dot{\theta} e^{i\theta} i \quad (3.87)$$

$$\ddot{\hat{r}} = \dot{r}e^{i\theta} + 2\dot{r}\dot{\theta}e^{i\theta}i - r\dot{\theta}^2e^{i\theta} + r\ddot{\theta}e^{i\theta}i \quad (3.88)$$

Nakon uprošćavanja i ubacivanja izraza za brzinu i ubrzanje u radijalni i tangencijalni oblik dobija se:

$$\dot{\hat{r}} = [\dot{r}]e^{i\theta} + [r\dot{\theta}]e^{i(\theta+\frac{\pi}{2})} \quad (3.89)$$

$$\ddot{\hat{r}} = [\dot{r} - r\dot{\theta}^2]e^{i\theta} + [2\dot{r}\dot{\theta} + r\ddot{\theta}]e^{i(\theta+\frac{\pi}{2})} \quad (3.90)$$

Izrazi za radijalnu i tangencijalnu brzinu i ubrzanje mogu se predstaviti na sledeći način:

$$v_u = \dot{r} \quad (3.91)$$

$$v_w = r\dot{\theta} \quad (3.92)$$

$$a_u = \dot{r} - r\dot{\theta}^2 \quad (3.93)$$

$$a_w = 2\dot{r}\dot{\theta} + r\ddot{\theta} \quad (3.94)$$

Na osnovu ovih četiri jednačina iznad, mogu se izvesti odgovarajuće veze između radijalne i tangencijalne brzine i ubrzanja robota:

$$a_u = \dot{v}_u - v_w\dot{\theta} \quad (3.95)$$

$$a_w = \dot{v}_w - v_u\dot{\theta} \quad (3.96)$$

Sledeći korak je pisanje Drugog Njutnovog zakona kretanja u robotskom koordinatnom sistemu u nalaženje veza između sila, obrtnog momenta i ubrzanja. Kod mobilnog robota s diferencijalnim pogonom postoje dva tipa kretanja : translacija u radijalnom (u) i tangencijalnom (w) pravcu i rotacija oko vertikalne ose u centru mase robota.

$$\sum F_u = ma_u$$

$$ma_u = F_{u_l} + F_{u_r} \quad (3.97)$$

$$\sum F_w = ma_w$$

$$ma_w = F_{w_l} - F_{w_r} \quad (3.98)$$

$$\sum M_C = I_C\ddot{\theta}$$

$$I_C\ddot{\theta} = (F_{w_r} - F_{w_l})d + (F_{u_r} - F_{u_l})r_c \quad (3.99)$$

Zamenom izraza za ubrzanje iz jednačina (3.95) i (3.96) u jednačinama (3.97) i (3.98) dobija se:

$$\dot{v}_u = v_w\dot{\theta} + \frac{F_{u_l} + F_{u_r}}{m} \quad (3.100)$$

$$\dot{v}_w = -v_u\dot{\theta} + \frac{F_{w_l} - F_{w_r}}{m} \quad (3.101)$$

$$\ddot{\theta} = \frac{(F_{wr} - F_{wl})d}{I_C} + \frac{(F_{ur} - F_{ul})r_c}{I_C} \quad (3.102)$$

Prve dve jednačine iznad pokazuju ubrzanje robota u zavisnosti od sila koje deluju kao i brzine.

Jednačine (3.100), (3.101) i (3.102) su osnovne jednačine dinamike diferencijalnog pogona mobilnog robota izvedene iz Njutnovog metoda dinamike.

Ograničenje klizanja u pravcu longituda (čisto kotrljanje) i ograničenje bočnog-lateralnog klizanja (nema klizanja) stvara nezavisnost između longitudinalne, lateralne i ugaone brzine i uprošćava jednačine dinamike.

Usled ograničenja bočnog klizanja, lateralna brzina centralne tačke I u robotskom koordinatnom sistemu je jednaka nuli, što je definisano jednačinom:

$$-\dot{x} \sin \theta + \dot{y} \cos \theta = 0$$

Korišćenjem rotacione matrice $\mathbf{R}(\theta)$, nalazimo brzinu centra mase u inercijalnom koordinatnom sistemu:

$$\mathbf{v}_C = \begin{bmatrix} \dot{x}_C \\ \dot{y}_C \end{bmatrix} = \begin{bmatrix} \cos \theta & -\sin \theta \\ \sin \theta & \cos \theta \end{bmatrix} \begin{bmatrix} v_u \\ v_w \end{bmatrix} \quad (3.103)$$

gde je :

$$\dot{x}_C = v_u \cos \theta - v_w \sin \theta \quad (3.104)$$

$$\dot{y}_C = v_u \sin \theta + v_w \cos \theta \quad (3.105)$$

Takođe korišćenjem jednačina (3.29) i (3.30) može se izračunati brzina centralne tačke I u inercijalnom koordinatnom sistemu.

Ubacivanjem jednačina (3.29) i (3.30) u jednačine (3.104) i (3.105) dobija se:

$$\dot{x} = v_u \cos \theta - v_w \sin \theta + d\dot{\theta} \sin \theta \quad (3.106)$$

$$\dot{y} = v_u \sin \theta + v_w \cos \theta - d\dot{\theta} \cos \theta \quad (3.107)$$

Kada se ove jednačine pomnože sa $-\sin \theta$ i $\cos \theta$, a zatim sabere dobija se:

$$-\dot{x} \sin \theta + \dot{y} \cos \theta = v_w - d\dot{\theta} \quad (3.108)$$

Odavde je lateralna brzina centralne tačke

$$v_w = d\dot{\theta} \quad (3.109)$$

A lateralno ubrzanje centralne tačke je:

$$\dot{v}_w = d\ddot{\theta} \quad (3.110)$$

Kombinacijom jednačina (3.110) i (3.101) dobijamo sledeće:

$$d\ddot{\theta} = -v_u \dot{\theta} + \frac{F_{w_l} + F_{w_r}}{m} \quad (3.111)$$

$$m(d\ddot{\theta} + v_u \dot{\theta}) = F_{w_l} + F_{w_r} \quad (3.112)$$

Kombinacijom jednačina (3.109) i (3.101) dobijamo:

$$\dot{v}_u = d\dot{\theta}^2 + \frac{F_{u_r} + F_{u_l}}{m} \quad (3.113)$$

Kombinacijom jednačina (3.112) i (3.102) i rešenja za $\ddot{\theta}$ imamo:

$$\ddot{\theta} = \frac{(F_{u_r} - F_{u_l})r_c}{md^2 + I_C} - \frac{mdv_u \dot{\theta}}{md^2 + I_C} \quad (3.114)$$

Ove dve jednačine iznad su jednačine dinamike robota uzimajući u obzir neholonomska ograničenja. Ove jednačine mogu jednostavno da se transformišu u formu matrice koristeći iste zapise i matrice kao kod Lagranžeovog metoda.

$$(md^2 + I_C)\ddot{\theta} + mdv\dot{\theta} = \frac{(\tau_R - \tau_L)r_c}{r} \quad (3.115)$$

$$m\dot{v} - md\dot{\theta}^2 = \frac{(\tau_R + \tau_L)}{r} \quad (3.116)$$

$$md^2 + I_C = I$$

Matrica formirana od ovih dveju jednačina je predstavljena pomoću sledeće jednačine:

$$\begin{bmatrix} m & 0 \\ 0 & I \end{bmatrix} \begin{bmatrix} \dot{v} \\ \dot{\theta} \end{bmatrix} + \begin{bmatrix} 0 & -md\dot{\theta} \\ md\dot{\theta} & 0 \end{bmatrix} \begin{bmatrix} v \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} \frac{1}{r} & \frac{1}{r} \\ \frac{r_c}{r} & \frac{-r_c}{r} \end{bmatrix} \begin{bmatrix} \tau_R \\ \tau_L \end{bmatrix} \quad (3.117)$$

Kao što se može videti iz jednačine iznad, iz oba metoda može se dobiti ista jednačina dinamike mobilnog robota. Koristeći jednačine (3.110) i (3.112) kao i jednačina direktne kinematike (3.8) i (3.9) možemo jednostavno napisati opšte jednačine dinamike u zavisnosti od obrtnog momenta aktuatora, rotacione brzine točkova i geometrijskih parametara. Diferenciranjem jednačina direktne kinematike (3.8) i (3.9) dobijamo:

$$\dot{v} = \frac{\ddot{\phi}_R + \ddot{\phi}_L}{2} r \quad (3.118)$$

$$\ddot{\theta} = \frac{\ddot{\phi}_R - \ddot{\phi}_L}{2r_c} r \quad (3.119)$$

Ubacivanjem jednačina (3.111) i (3.8) u jednačinu (3.112) dobijamo sledeći izraz:

$$\frac{\ddot{\phi}_R - \ddot{\phi}_L}{2r_c} r = \frac{(F_{u_r} - F_{u_l})r_c}{I} - \frac{md}{I} \left(\frac{\dot{\phi}_R + \dot{\phi}_L}{2} r_c \right) \left(\frac{\dot{\phi}_R - \dot{\phi}_L}{2r} r_c \right) \quad (3.120)$$

Ubacivanjem jednačina (3.110) i (3.9) u jednačinu (3.109) dobijamo:

$$\frac{\ddot{\varphi}_R + \ddot{\varphi}_L}{2} r = d \left(\frac{\dot{\varphi}_R - \dot{\varphi}_L}{2r_c} r \right)^2 + \frac{F_{u_r} + F_{u_l}}{m} \quad (3.121)$$

Iz jednačina (3.112) i (3.96) dobija se:

$$F_{u_r} - F_{u_l} = \frac{rI}{2r_c^2} (\ddot{\varphi}_R - \ddot{\varphi}_L) + \frac{md}{r_c} \left(\frac{\dot{\varphi}_R + \dot{\varphi}_L}{2} r \right) \left(\frac{\dot{\varphi}_R - \dot{\varphi}_L}{2r_c} r \right) \quad (3.122)$$

$$F_{u_r} + F_{u_l} = \frac{mr}{2} (\ddot{\varphi}_R + \ddot{\varphi}_L) - dm \left(\frac{\dot{\varphi}_R - \dot{\varphi}_L}{2r_c} r \right)^2 \quad (3.123)$$

Dodavanjem i uprošćavanjem jednačina iznad dobija se:

$$F_{u_r} = \left(\frac{rI}{4r_c^2} + \frac{mr}{4} \right) \ddot{\varphi}_R + \left(-\frac{rI}{4r_c^2} + \frac{mr}{4} \right) \ddot{\varphi}_L - \left(\frac{mdr^2}{4r_c^2} \right) \dot{\varphi}_L^2 + \left(\frac{mdr^2}{4r_c} \right) \dot{\varphi}_R \dot{\varphi}_L \quad (3.124)$$

$$F_{u_l} = \left(\frac{rI}{4r_c^2} + \frac{mr}{4} \right) \ddot{\varphi}_L + \left(-\frac{rI}{4r_c^2} + \frac{mr}{4} \right) \ddot{\varphi}_R - \left(\frac{mdr^2}{4r_c^2} \right) \dot{\varphi}_R^2 + \left(\frac{mdr^2}{4r_c^2} \right) \dot{\varphi}_R \dot{\varphi}_L \quad (3.125)$$

Pod pretpostavkom da su ulazi sistema obrtni momenti DC motora, jednačine dinamike su:

$$\begin{aligned} \tau_R = & \left(\frac{r^2 I}{4r_c^2} + \frac{mr^2}{4} \right) \ddot{\varphi}_R + \left(-\frac{r^2 I}{4r_c^2} + \frac{mr^2}{4} \right) \ddot{\varphi}_L - \left(\frac{mdr^3}{4r_c^2} \right) \dot{\varphi}_L^2 \\ & + \left(\frac{mdr^3}{4r_c^2} \right) \dot{\varphi}_R \dot{\varphi}_L \end{aligned} \quad (3.126)$$

$$\begin{aligned} \tau_L = & \left(\frac{r^2 I}{4r_c^2} + \frac{mr^2}{4} \right) \ddot{\varphi}_L + \left(-\frac{r^2 I}{4r_c^2} + \frac{mr^2}{4} \right) \ddot{\varphi}_R - \left(\frac{mdr^3}{4r_c^2} \right) \dot{\varphi}_R^2 \\ & + \left(\frac{mdr^3}{4r_c^2} \right) \dot{\varphi}_R \dot{\varphi}_L \end{aligned} \quad (3.127)$$

Sledeći parametri su definisani kako bi pojednostavili ostatak računanja:

$$A = \frac{r^2 I}{4r_c^2} + \frac{mr^2}{4} \quad (3.128)$$

$$B = -\frac{r^2 I}{4r_c^2} + \frac{mr^2}{4} \quad (3.129)$$

$$C = \frac{m d r^3}{4 r_c^2} \quad (3.130)$$

Prema tome, jednačine dinamike možemo zapisati na sledeći način:

$$\tau_R = A \ddot{\phi}_R + B \ddot{\phi}_L - C \dot{\phi}_L^2 + C \dot{\phi}_R \dot{\phi}_L \quad (3.131)$$

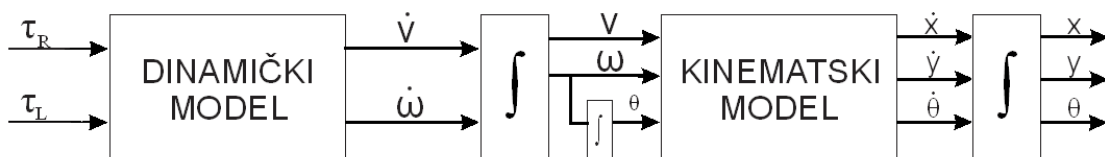
$$\tau_L = A \dot{\phi}_L + B \ddot{\phi}_R - C \dot{\phi}_R^2 + C \dot{\phi}_R \dot{\phi}_L \quad (3.132)$$

Jednačine (3.131) i (3.132) su jednačine dinamike koje povezuju obrtne momente, rotacionu brzinu točkova i rotaciono ubrzanje. Za simulaciju i razvoj kontrolera mogu se koristiti ili jednačine (3.131) i (3.132) ili (3.117) iz Njutnovog metoda ili jednačina (3.84) iz Lagranževog metoda. Za potrebe ove teze korišćena je (3.84) za simulaciju sistema koji će biti opisan u sledećem odeljku. Isti metod dinamičkog modeliranja je korišćen u referencama [30] [39].

3.4. SIMULACIJA SISTEMA

U cilju testiranja razvijenog upravljanja mobilnim robotom pre testiranja na realnom robotu, u ovoj tezi je razvijen simulacioni model.

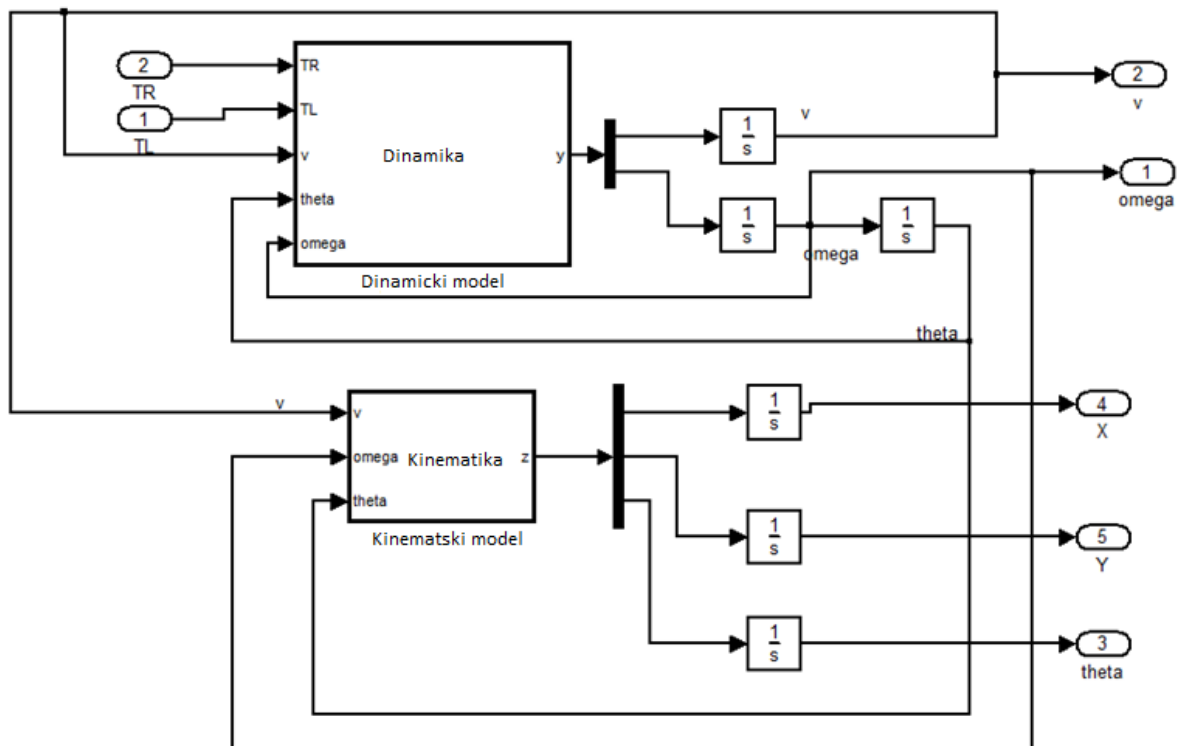
Strukturna šema mehaničkog modela mobilnog robota koja uključuje modele kinematike i dinamike je prikazana na Slici 3.7. Na slici 3.8 je prikazan simulacioni model kojim simuliramo mehanički model bez uključivanja efekata aktuatora. Jednostavno povežemo različite obrtne momente kao izlaze desnog i levog aktuatora točka za mehanički model i analiziramo kretanje robota pod uticajem ovih obrtnih momenata.



Slika 3.7: Strukturna šema dinamičkog i kinematskog modela

Jednačine (3.81) do (3.84) za dinamički model i jednačina (3.66) za kinematski model su implementirane u formi funkcija da bi simulirale mehanički model robota kao što je prikazano na slici iznad. Dinamički model i kinematski model funkcija su uključene u dodatku A.

Nakon izgradnje modela treba definisati parametre robota i uraditi neke testove za proveru performansi modela. Parametri robota korišćeni za simulaciju modela prikazani su u Tabeli 3.1:



Slika 3.8: Simulacioni model dinamike i kinematike mobilnog robota

Tabela 3.1: Parametri robota korišćeni za simulaciju

| Ime parametra | Vrednost parametra | Opis |
|---------------|---------------------------|---|
| m | 3.6 (KG) | Težina robota |
| r_c | 0.2025 (m) | Rastojanje između pogonskog točka i ose simetrije |
| d | 0.05 (m) | Rastojanje između centra mase i ose pogonskog točka |
| r | 0.05 (m) | Poluprečnik točka |
| I_c | 0.09 (KG*m ²) | Masa momenta inercije oko centra mase |

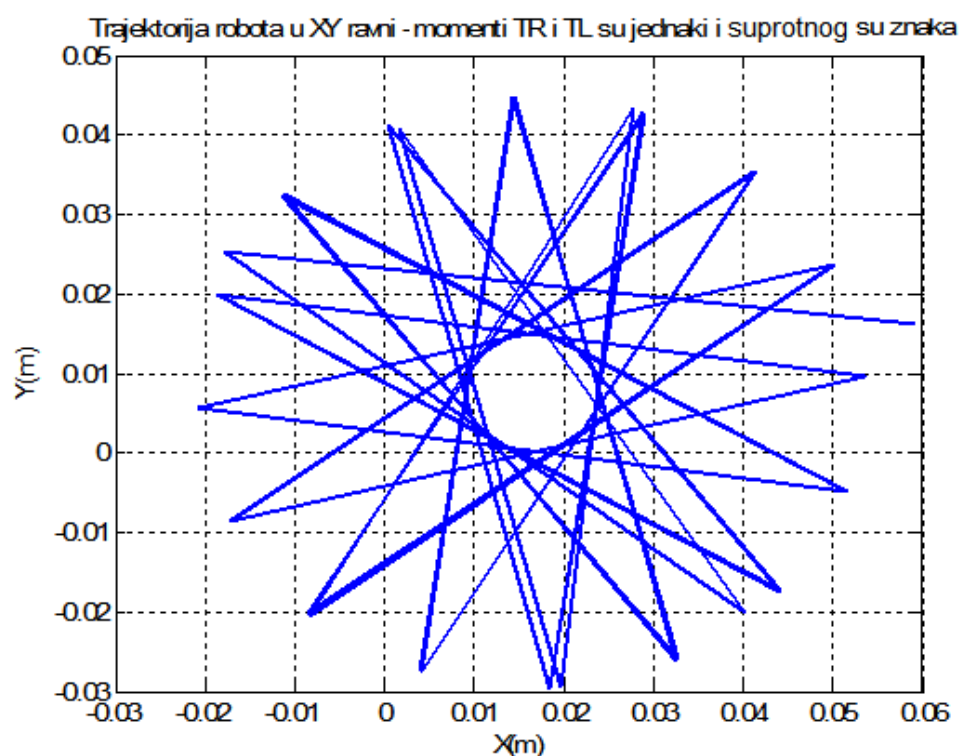
Navedeni parametri se mogu uzeti iz liste podataka o DaNI robotu, koji je razvijen od strane National Instruments-a,[113]. Dimenzije robota su 405mm × 368mm × 150 mm.

Nakon definisanja parametara robota, rade se testovi za proveru modela.

Četiri moguća slučaja za ulazne signale su:

- ulazni pogonski momenti τ_R i τ_L su jednaki ali suprotnog znaka- robot će se okretati oko svoje ose.
- ulazni pogonski momenti τ_R i τ_L su jednaki i istog znaka- robot će se kretati pravolinijski.
- ulazni pogonski moment τ_R je manji od ulaznog pogonskog moment τ_L istog znaka- robot će se kretati udesno.
- ulazni pogonski moment τ_R je je manji od ulaznog pogonskog moment τ_L istog znaka- robot će se kretati ulevo.

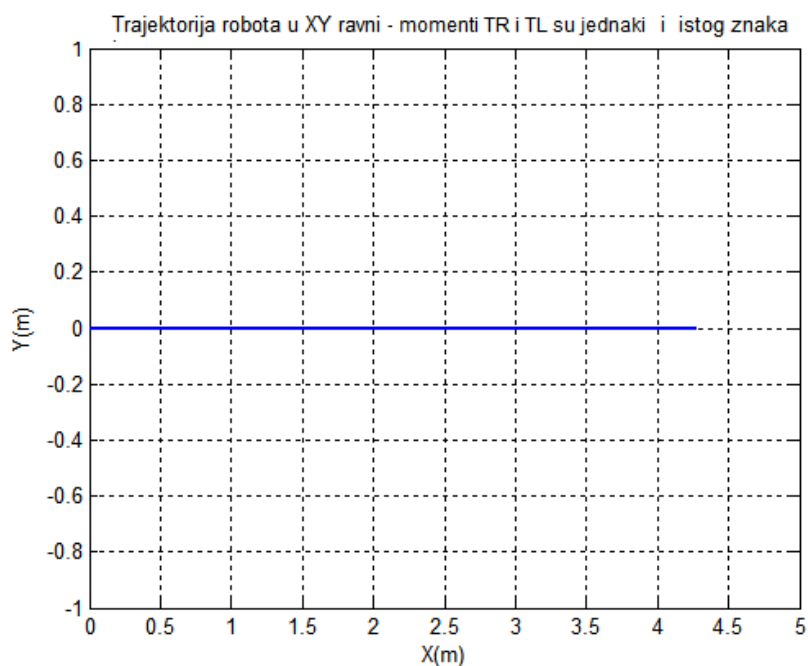
Simulirano kretanje robota kao posledica prvog slučaja je prikazano na Slici 3.9.



Slika 3.9: Kretanje robota kao posledica istih pogonski momenata τ_R i τ_L ali suprotnog znaka

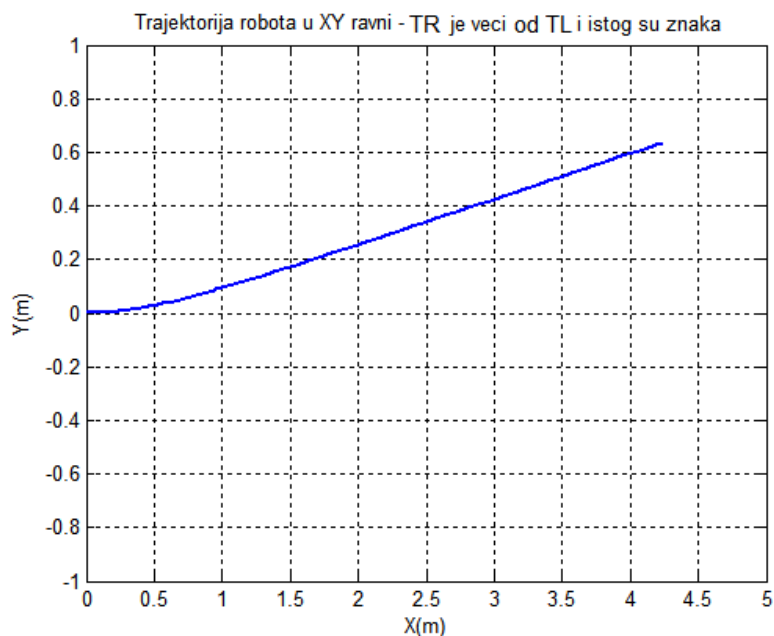
Kao što se može videti, robot će pri kretanju napraviti neku vrstu kruga oko centralne tačke kada obrtni momenta motora imaju jednake suprotne znakove. Robot bi trebalo da rotira oko svoje ose kao posledica takvog ulaza. Razlog zbog koga pravi “krug“ i rotira oko centralne tačke je odstupanje centra mase od centra rotacije.

Drugi test za proveru modela je dodeljivanje jednakih ulaznih pogonskih momenta τ_R i τ_L koji su istog znaka.

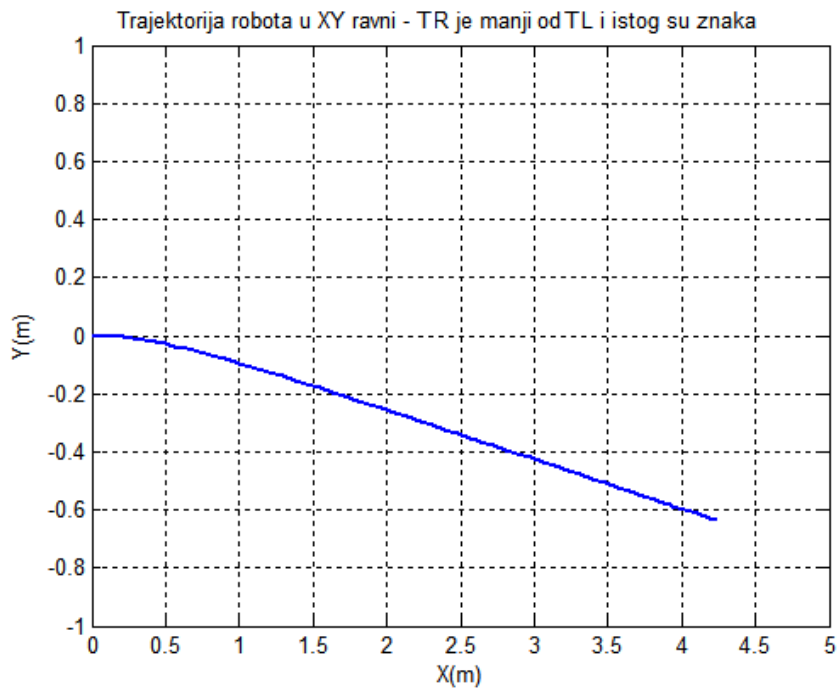


Slika 3.10: Kretanje robota kao posledica istih pogonskih momenata τ_R i τ_L koji su i istog znaka

Treći test za proveru modela je dodeljivanje istih znakova ali nejednakih ulaznih obrtnih momenata robotu i posmatranje kretanje robota. Na slikama 3.11 i 3.12 prikazano je kretanje robota kao odgovor na isti znak ali nejednakih ulaznih obrtnih momenata.



Slika 3.11: Kretanje robota u slučaju ulaznih pogonskih momenata istog znaka, ali τ_R veći od ulaznog pogonskog momenta τ_L - robot se kreće ulevo.



Slika 3.12: Kretanje robota za slučaj ulaznog pogonskog momenta τ_R istog znaka sa τ_L , ali manjeg od ulaznog pogonskog momenta τ_L - robot se kreće udesno.

Gledajući slike iznad može se videti da rezultati simulacije odgovaraju kretanju diferencijalnog pogona mobilnog robota i da se izvedeni model može koristiti za razvoj upravljanje o čemu će biti reči u narednom poglavlju.

4. PROJEKTOVANJE KONTROLERA ZA PRAĆENJE ČOVEKA

U ovom poglavlju razmatra se problem dizajniranja kontrolera za neholonomske mobilne robote u cilju praćenja čoveka na osnovu vizuelne informacije.

Praćenje pokretnih objekata, uključujući i čoveka je važan proces u različitim aplikacijama kompijuterske vizije i robotike. Praćenje pokretnog objekta je jedan od najznačajnijih zadataka za autonomne mobilne robote. Definiše se kao sposobnost robota da podesi svoju poziciju i orijentaciju prema kretanju pokretnog objekta u cilju smanjenja greške praćenja. U predhodnim godinama razvijene su različite šeme upravljanja za navigaciju i praćenje mobilnim robotom.

Metod veštačkog potencijalnog polja za praćenje pokretne mete je uveden u upotrebu 1986. godine od strane Khaitb-a [43]. Kod ovog metoda robot je predstavljen kao čestica u modifikovanom prostoru q koja se kreće pod uticajem veštačkog potencijala nastalog pod uticajem vrednosti pozicije mete q_{goal} i udaljenosti od objekata koje predstavljaju prepreku kretanju robota. Nedostatak ove tehnike je što robot često ostane “zarobljen” na nekom od lokalnih minimuma [44]. Zadnjih godina veliki napor je uloženi u razvoj inteligentnih zakona upravljanja. Metode veštačke inteligencije se mogu primeniti za različite probleme sa kojima se srećemo kod mobilne robotike [45]. Ove metode se često koriste zbog svoje jednostavnosti, jer su zasnovane na ljudskom rezonovanju i heurističkom znanju i ne zahtevaju matematički model. Razvijeno je nekoliko kontrolera zasnovanih na fazi logici [46][47], neuronskim mrežama [39,48] kao i na genetskim algoritmima [45][49]. Glavni nedostatak ovih tehnika je nedostatak preciznog modela i zbog toga što su računarski zahtevne.

Pored navedenih tehnika u upotrebi su i tehnike nelinearnog upravljanja mobilnim robotom kao što su upravljanje po kliznom režimu (*eng.* Sliding mode control) [50][51] i nelinearni kontroleri zasnovani na metodama Ljapunova [52][53].

Upravljanje neholonomskim mobilnim robotom možemo podeliti takođe i na sledeća dva metoda upravljanja: upravljanje na osnovu kinematskog modela i upravljanje na osnovu dinamičkog modela mobilnog robota. U ranijim istraživanjima, veliki broj istraživača je bio fokusiran samo na kinematsko upravljanje mobilnim robotom. Međutim, zadnjih godina veliki broj istraživanja je rađen na razvoju aplikacija na osnovu dinamičkog upravljanja [54].

Kinematski bekstepping kontroler (*eng.* Kinematic backstepping controller) za upravljanje neholonomskim mobilnim robotom koji zanemaruje dinamiku vozila i koji koristi Ljapunove funkcije prvi put je predstavio Kanayama [55] 1991. godine i od tada ga koriste mnogi istraživači u

ovom polju. Izračunavanje željene linearne i ugaone brzine $v_c = (v, w)$ se vrši na osnovu upravljačkog zakona predloženog u [55].

Kontroleri zasnovani na dinamičkom modelu mogu se svrstati u četiri grupe, u zavisnosti od toga kako se tretira dinamika sistema [56]: adaptivni kontroleri, robusni kontroleri, robusni adaptivni kontroleri i kontroleri sa linearizovanom povratnom spregom .

Adaptivni kontroleri su adaptivne ekstenzije kinematskim bekstepping kontrolerima koji mogu da se izbore sa nepoznatim parametrima [32],[57],[58].

Robusni kontroleri su dizajnirani da se izbore sa neodređenim parametrima i poremećajima u sistemu [59],[60].

Robusni adaptivni kontroleri imaju i robusne i adaptivne karakteristike i najčešće koriste kombinaciju inteligentnih kontrolera i nelinearno upravljanje [56].

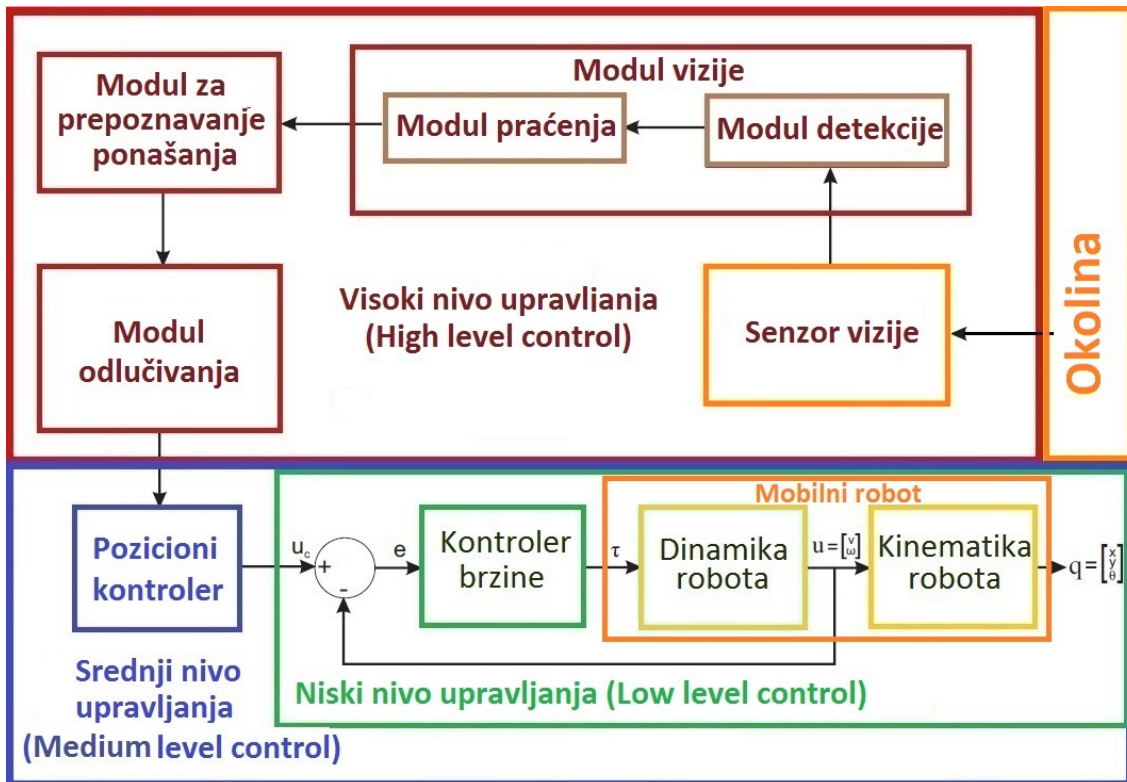
Upravljanje na osnovu linearizovane povratne sprege je prikazano u [61]. Takođe sve je više metoda koje obuhvataju i pozicioni kontroler i upravljanje brzinama točkova [62].

4.1. RAZVOJ HIJERARHIJSKE STRUKTURE UPRAVLJANJA MOBILNIM ROBOTOM SA DIFERENCIJALNIM POGONOM ZA PRAĆENJE LJUDI

Kao što je već, u uvodu, napomenuto za robote koji pružaju podršku osobama u izvršavanju raznih zadataka, kao što su roboti koji pomažu osobama sa invaliditetom, roboti-pratioci starijih ljudi ili mobilni roboti koji asistiraju ljudima u inspekciji kontaminiranih/opasnih okruženja, ključni uslov za generisanje adekvatnog ulaza u upravljački sistem, pored detekcije i praćenja ljudi, je prepoznavanje i predikcija ponašanja ljudi, kao i modul odlučivanja koji na osnovu ponašanja čoveka donosi odluku o upravljanju mobilnim robotom.

U hijerarhijskoj strukturi upravljanja (Slika 4.1) razvijenoj u ovoj tezi moduli za detekciju, praćenje i predikciju i prepoznavanje ponašanja čoveka kao i modul odlučivanja su na visokom nivou strukture upravljanja (*eng.* high-level control). Visoki nivo upravljanja je na slici 4.1 označen crevenom bojom. Upravljanje pozicijom i orijentacijom mobilnog robota se svrstava u srednji nivo upravljanja (*eng.* medium-level) i označeno je plavom bojom na pomenutoj slici. Upravljački sistem niskog nivoa (*eng.* low-level control), na slici 4.1 je označen zelenom bojom i sastoji se od multivarijabilnog PD kontrolera koji je odgovoran za konvertovanje brzina na izlazu pozicionog kontrolera u obrtni moment točkova robota. Ovaj kontroler je potreban da bi se osiguralo praćenje ulazne referentne brzine od strane robota. Senzor vizije i mobilni robot (koji je se može opisati kinematskim i dinamičkim jednačinama) su označeni narandžastom bojom i odvojeni od ostatka sistema zbog njihove interakcije sa okolinom. Naime senzor vizije pruža informacije o prostoru koji okružuje robota, dok se robot kreće u tom istom prostoru. U ovom poglavlju opisani su razvoj i

testiranje modula niskog i srednjeg nivoa sistema upravljanja. Upravljački nivo visokog nivoa je zbog svoje kompleksnosti obrađen u narednim poglavljima.



Slika 4.1 Hijerarhijska struktura upravljanja mobilnim robotom

4.2. UPRAVLJAČKI SISTEM NISKO NIVOVA –KONTROLER BRZINE

Upravljački sistem niskog nivoa sastoji se od multivarijabilnog PD kontrolera koji je odgovoran za konvertovanje brzina na izlazu pozicionog kontrolera u obrtni moment točkova robota. Obrtni momenti su podešeni prema sledećoj jednačini:

$$\boldsymbol{\tau} = \begin{bmatrix} \tau_R \\ \tau_L \end{bmatrix} = \mathbf{B}^{-1} \begin{bmatrix} g_1(t) & 0 \\ 0 & g_2(t) \end{bmatrix} \begin{bmatrix} e_v(t) \\ e_\omega(t) \end{bmatrix} \quad (4.1)$$

$$g_1(t) = K_{pv} e_v(t) + K_{dv} \frac{de_v(t)}{dt}$$

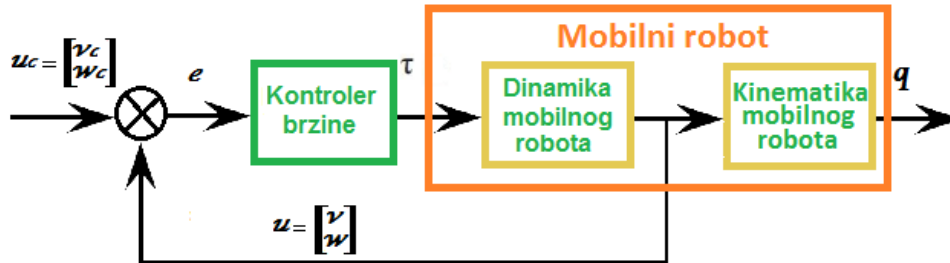
$$g_2(t) = K_{p\omega} e_\omega(t) + K_{d\omega} S \frac{de_\omega(t)}{dt}$$

$$\mathbf{B}^{-1} = \begin{bmatrix} \frac{r}{2} & \frac{r}{2r_c} \\ r & r \\ \frac{r}{2} & -\frac{r}{2r_c} \end{bmatrix}$$

gde je $e_v(t)$ greška linearne brzine, a $e_\omega(t)$ greška ugaone brzine mobilnog robota.

$$e = \begin{bmatrix} e_v(t) \\ e_\omega(t) \end{bmatrix} = \begin{bmatrix} u_1(t) - v(t) \\ u_2(t) - \omega(t) \end{bmatrix} \quad (4.2)$$

Strukturna šema ovakve upravljačke strukture je prikazana na slici 4.2.

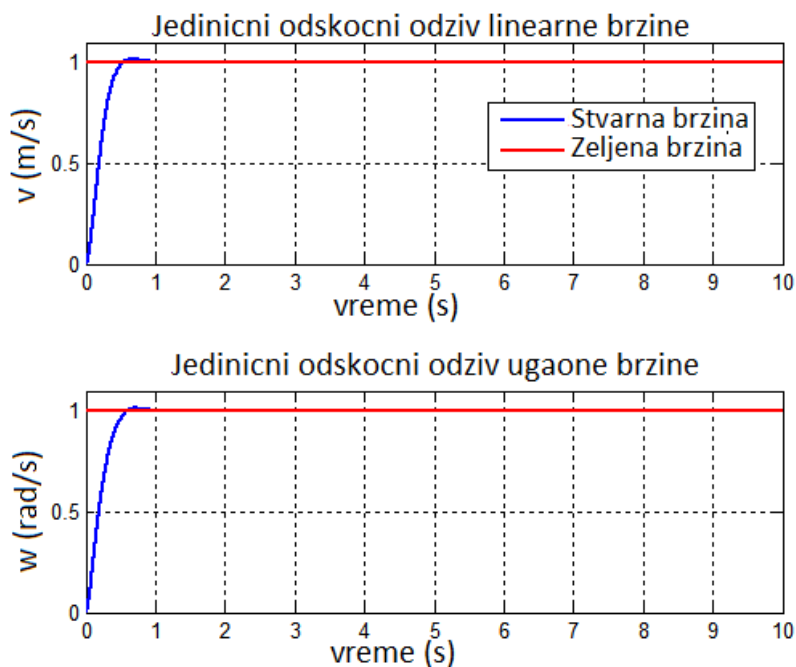


Slika 4.2. Kontroler brzine

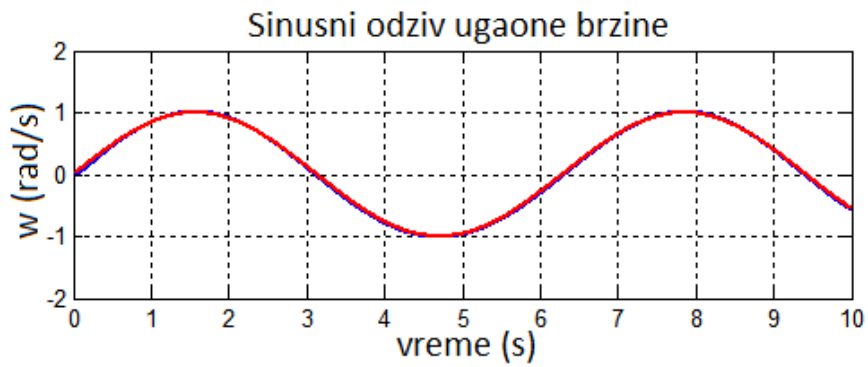
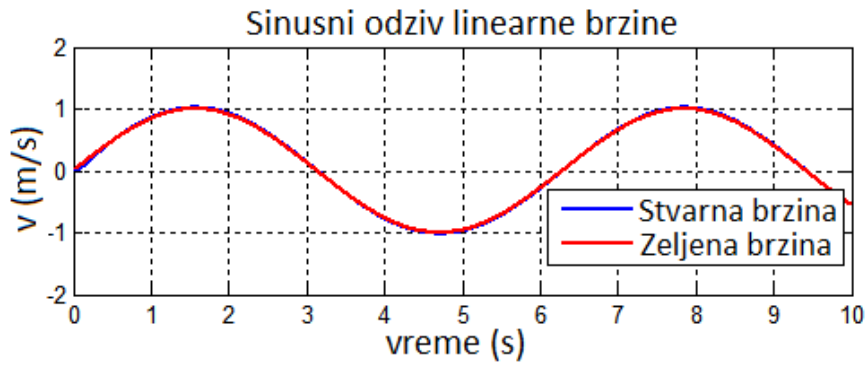
Ideja je da se vrši upravljanje s povratnom spregom na osnovu dinamičkog modela robotskog sistema. Dinamički model robotskog sistema koji je opisan jednačinom (3.82) ima dva ulaza (τ_R , τ_L) i dva izlaza (v, ω).

Kvalitet upravljanja sa povratnom spregom testira se preko praćenja zadatih linearnih i ugaonih brzina robota. Rezultati simulacije prikazani su na slikama 4.3, 4.4 i 4.5, i može se videti da predloženi kontroler brzine uspešno prati zadate brzine. Parametri kontrolera su optimizovani uz pomoć *optimizacije rojevima čestica* (eng. Particle Swarm Optimization), koja je detaljnije objašnjena u poglavlju 4.4.

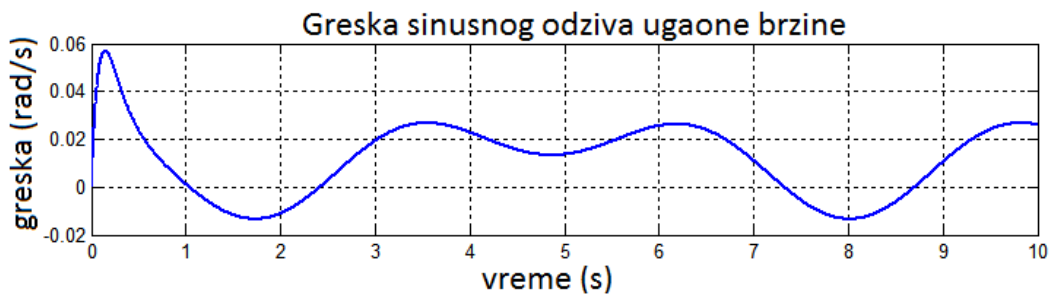
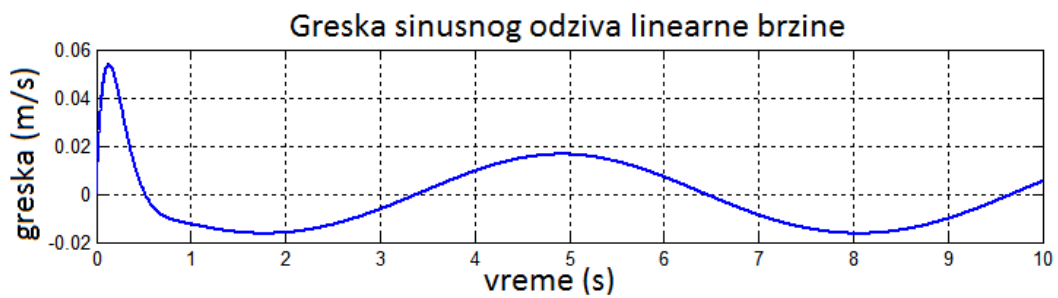
Vrednosti parametara kontrolera su sledeće: $K_{pv}=218$, $K_{dv}=45$, $K_{pw}=181$, $K_{dw}=28$.



Slika 4.3 . Odskočni odziv linearne i ugaone brzine (eng. Step response)



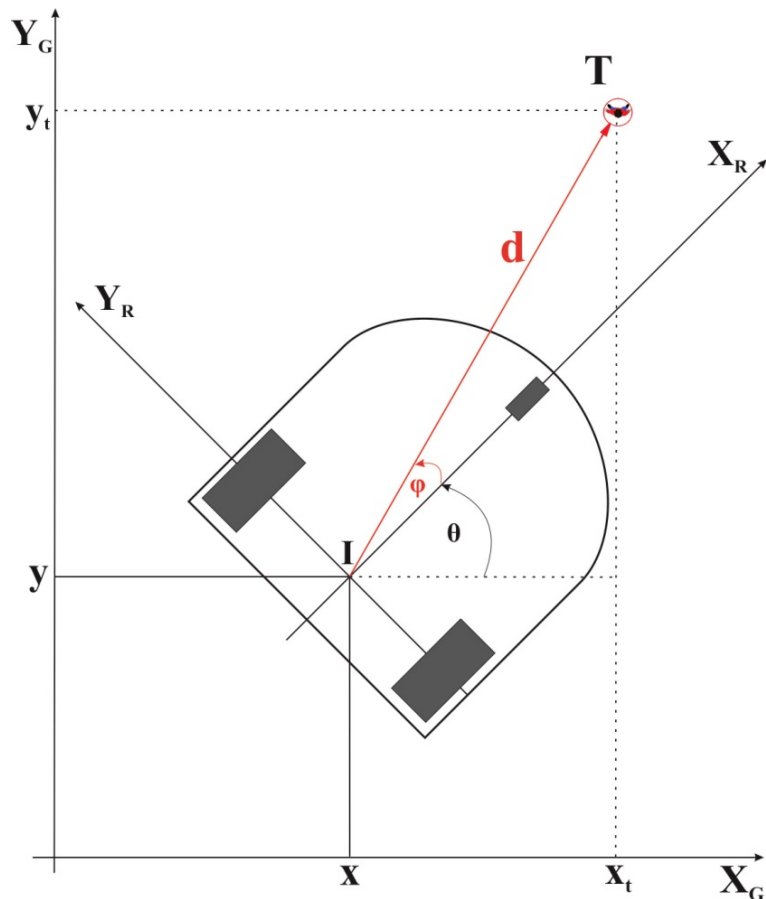
Slika 4.4. Sinusni odziv linearne i ugaone brzine (*eng.* Sine response)



Slika 4.5. Greška linearne i ugaone brzine (*eng.* Sine response error)

4.3. UPRAVLJAČKI SISTEM SREDNJEG NIVOVA

Upravljački sistem srednjeg nivoa je projektovan tako da pokreće robota s njegove trenutne pozicije $\mathbf{q} = [x \ y \ \theta]^T$ prema željenoj poziciji $\mathbf{q}_t = [x_t \ y_t \ \theta_t]^T$. Cilj je naći upravljački signal $\mathbf{u}_c = [v_c \ w_c]$ tako da robot prati čoveka, tj. da dostigne željenu poziciju u konačnom vremenskom intervalu. Na isti način pozicije robota i čoveka kog treba pratiti mobilnim robotom se mogu predstaviti u polarnom koordinatnom sistemu (Slika 4.8), pri čemu rastojanje između robota i čoveka, d mora biti veće od nule ($d > 0$).



Slika 4.6. Mobilni robot prati čoveka

Pozicija robota i čoveka u globalnom koordinatnom sistemu i rastojanje d između njih, mogu se povezati sledećim jednačinama:

$$d \cos \alpha = x_t - x \quad (4.3)$$

$$d \sin \alpha = y_t - y$$

dok odnos između pravca kretanja robota i čoveka, definišemo preko sledećih promenljivih:

$$\varphi = \alpha - \theta \quad (4.4)$$

Brzinu kretanja čoveka se ne uzima u obzir prilikom dizajniranja ovog kontrolera te je na osnovu jednačine (4.3):

$$\dot{d} \cos \alpha - d \dot{\alpha} \sin \alpha = -v \cos \theta \quad (4.5)$$

$$\dot{d} \sin \alpha + d \dot{\alpha} \cos \alpha = -v \sin \theta \quad (4.6)$$

$$\dot{\varphi} = \dot{\alpha} - w \quad (4.7)$$

Ukoliko se jednačina (4.5) pomnoži sa $\cos \alpha$, a jednačinu (4.6) $\sin \alpha$ sa dobija se:

$$\dot{d} \cos^2 \alpha - d \dot{\alpha} \sin \alpha \cos \alpha = -v \cos \theta \cos \alpha$$

$$\dot{d} \sin^2 \alpha + d \dot{\alpha} \cos \alpha \sin \alpha = -v \sin \theta \sin \alpha$$

i na osnovu adicijonih formula dobija se:

$$\dot{d} = -v \cos \varphi \quad (4.8)$$

Takođe, ukoliko se jednačina (4.5) pomnoži sa $-\sin \alpha$, a jednačinu (4.8) sa $\cos \alpha$ sa dobija se:

$$-\dot{d} \cos \alpha \sin \alpha + d \dot{\alpha} \sin^2 \alpha = v \cos \theta \sin \alpha$$

$$\dot{d} \sin \alpha \cos \alpha + d \dot{\alpha} \cos^2 \alpha = -v \sin \theta \cos \alpha$$

što je dalje

$$\dot{\alpha} = \frac{v \sin \varphi}{d}, d \neq 0 \quad (4.9)$$

Na osnovu (4.7) i (4.9) sledi

$$\dot{\varphi} = -w + \frac{v \sin \varphi}{d}, d \neq 0 \quad (4.10)$$

Kinematski model je sada predstavljen novim promenljivima stanja d, φ i α .

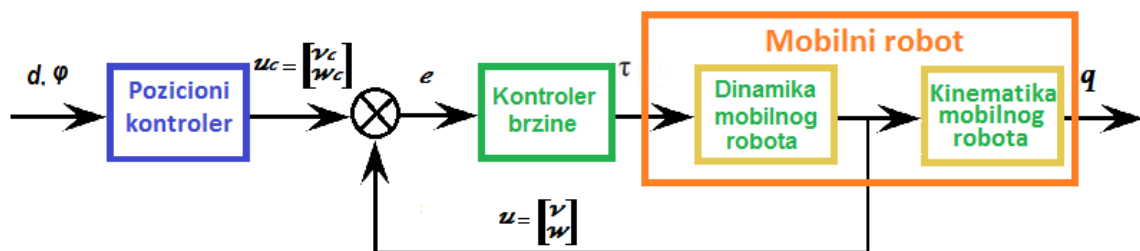
Originalni problem praćenja postaje problem regulacije sistema opisanog jednačinama (4.8), (4.9) i (4.10).

Predloženo kinematsko pravilo upravljanja je opisano na sledeći način:

$$\mathbf{u}_c = \begin{bmatrix} K_p d \cos \varphi \\ K_p \sin \varphi \cos \varphi + K_a \varphi \end{bmatrix} \quad (4.11)$$

$$K = (K_p, K_a)$$

Strukturna šema ovakve kontrolne strukture je prikazana na slici 4.7.



Slika 4.7. Pozicioni kontroler

4.3.1. Ljapunovljeve analize stabilnosti

Jedan od najčešće korišćenih metoda za analizu asimptotskog ponašanja zasnovan je na Ljapunovoj teoriji stabilnosti.

Jednostavna pozitivno definisana kvadratna forma Ljapunovljeve funkcije je:

$$V = V_1 + V_2 = \frac{1}{2}d^2 + \frac{1}{2}\varphi^2 \quad (4.12)$$

gde parametri V_1, V_2 predstavljaju polovinu kvadratne norme greške rastojanja d i greške orijentacije φ nastalih kao rezultat rastojanja robota između njegove trenutne i željene pozicije definisanih u globalnom koordinatnom sistemu.

Prema tome izvod predložene Ljapunovljeve funkcije je:

$$\dot{V} = \dot{V}_1 + \dot{V}_2 = d\dot{d} + \varphi\dot{\varphi} \quad (4.13)$$

Korišćenjem kinematskih jednačina (4.8) i (4.10), dobija se:

$$\dot{V} = d(-v\cos\varphi) + \varphi(-w + \frac{v\sin\varphi}{d}) \quad (4.14)$$

Zamenom v i ω sa $v(d, \varphi)$ i $\omega(d, \varphi)$ dobija se:

$$\dot{V} = d(-K_p d \cos^2 \varphi) + \varphi(-K_p \sin \varphi \cos \varphi - K_a \varphi + \frac{K_p d \cos \varphi \sin \varphi}{d}) \leq 0 \quad (4.15)$$

U jednačini (4.13) prvi član je

$$\dot{V}_1 = -K_p d^2 \cos^2 \varphi \leq 0, \text{ za } K_p > 0 \quad (4.16)$$

Kako je V_1 funkcija pozitivno definitna i \dot{V}_1 negativno definitna funkcija, tada po Ljapunovljevoj teoremi trivijalno rešenje sistema je asimptotski stabilno.

Slično za drugi član je:

$$\dot{V}_2 = \varphi(-K_p \sin \varphi \cos \varphi - K_a \varphi + \frac{K_p d \cos \varphi \sin \varphi}{d}) \leq 0 \quad (4.17)$$

$$\dot{V}_2 = -K_a \varphi^2 \leq 0, \text{ za } K_a > 0 \quad (4.18)$$

Takođe V_2 je funkcija pozitivno definitna i \dot{V}_2 negativno definitna funkcija, i sledi da je trivijalno rešenje sistema je asimptotski stabilno.

Konačno je:

$$\dot{V} = \dot{V}_1 + \dot{V}_2 = -K_p d^2 \cos^2 \varphi - K_a \varphi^2 \leq 0 \quad (4.19)$$

Primenom Barbalet'ove Leme [63], sledi da \dot{V} konvergira ka nuli kako vreme raste. Ovo implicira da vektor $[d, \varphi, \alpha]$ konvergira ka $[0, 0, \alpha]$.

Prema tome može se zaključiti da upravljačke veličine

$$\begin{aligned} v_c &= K_p d \cos \varphi \\ \omega_c &= K_p \sin \varphi \cos \varphi + K_a \varphi \end{aligned} \quad (4.20)$$

čine kretanje robota glatkim i stabilnim.

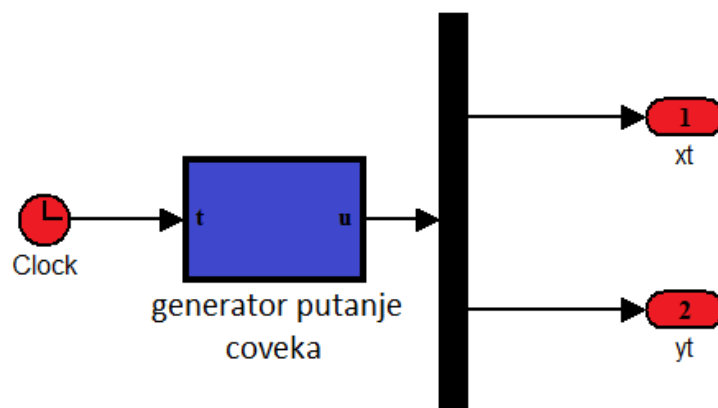
4.3.2 Simulacija pozicionog kontrolera

Da bi se testirao pozicioni kontroler, prvo je potrebno simulirati kretanje čoveka. Šema generatora željene putanje prikazana na slici 4.9 generiše željenu putanju kretanja čoveka. Putanja je predstavljena kao funkcija vremena:

$$x_t(t) = f(t), y_t(t) = g(t) \quad (4.21)$$

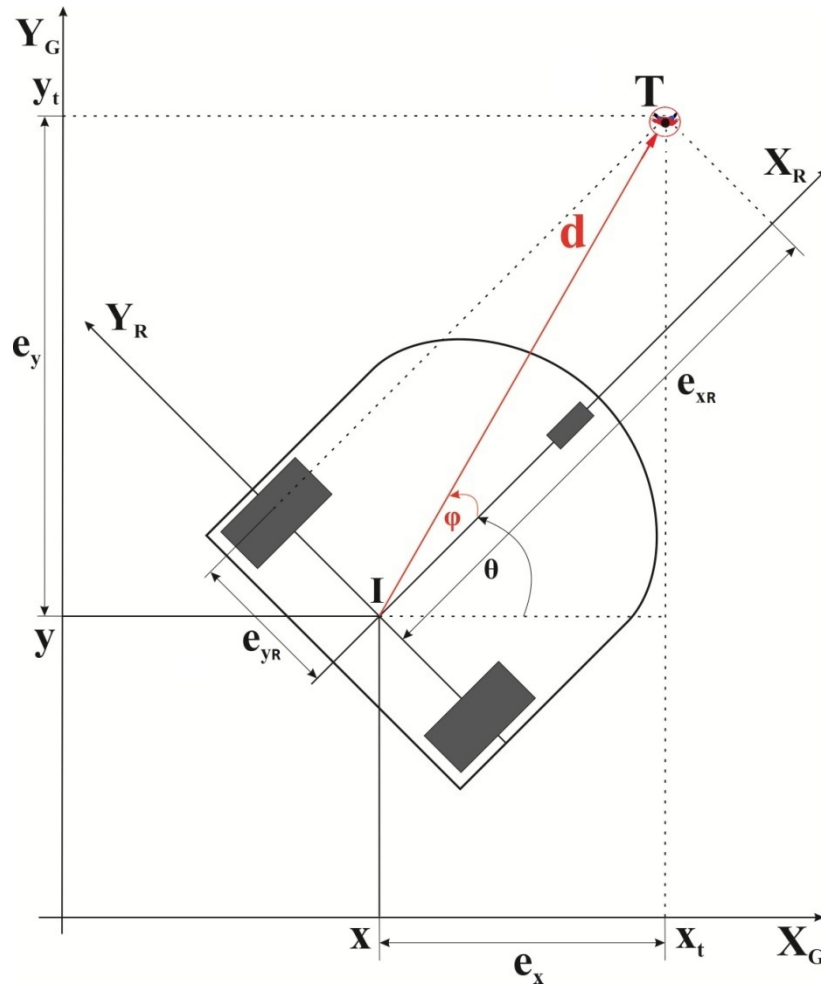
Brzina čoveka $v_t(t)$ se izračunava kao prvi izvod jednačine (4.21).

U cilju testiranja performansi upravljačke strukture nekoliko različitih tipova putanja je simulirano, pri čemu se čovek kreće konstantnom brzinom oko 1 m/s. Takođe je testiran i slučaj kad se čovek kreće promenljivom brzinom.



Slika 4.8. Šema generatora putanje čoveka

Transformacija se vrši na osnovu greške praćenja u globalnom i robotskom koordinatnom sistemu kao što je prikazano na slici 4.9.



Slika 4.9. Greške pozicije i orijentacije čoveka u odnosu na robot u globalnom i robotskom koordinatnom sistemu.

Greška pozicije u centru koordinatnog sistema povezanog sa mobilnom platformom ili lokalnim koordinatnim sistemom definiše se na sledeći način:

$$\mathbf{e}_p = \begin{bmatrix} e_{xR} \\ e_{yR} \end{bmatrix} = \begin{bmatrix} x_{HR} \\ y_{HR} \end{bmatrix} = \begin{bmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{bmatrix} \begin{bmatrix} e_x \\ e_y \end{bmatrix}$$

$$\begin{bmatrix} e_x \\ e_y \end{bmatrix} = \begin{bmatrix} x_t - x \\ y_t - y \end{bmatrix} \quad (4.22)$$

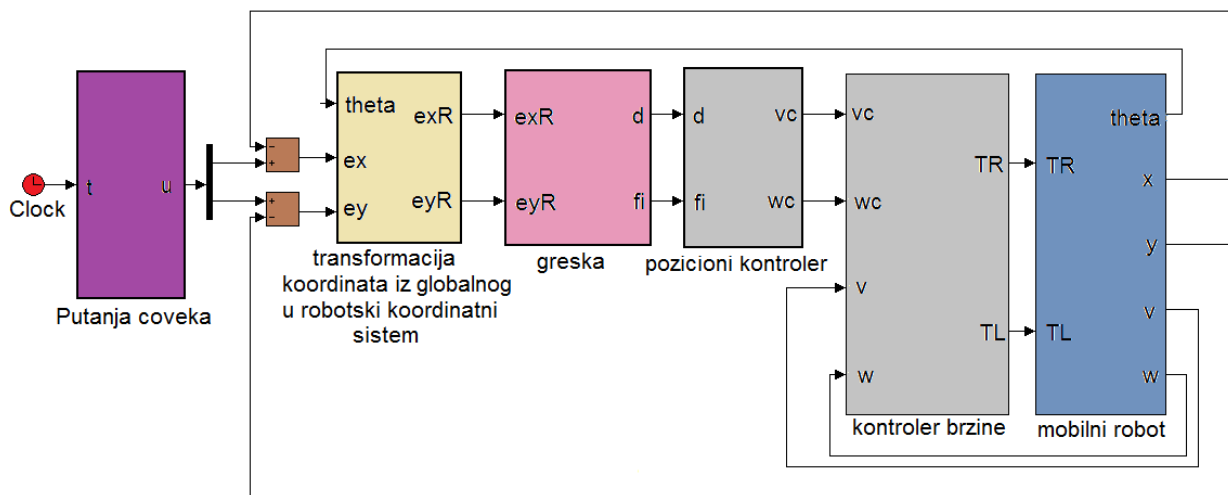
Gde je x_{HR} i y_{HR} pozicija čoveka u robotskom koordinatnom sistemu, tj. pozicija u kojoj treba da dođe robot da bi greška praćenja bila jednaka nuli.

Sada se greška rastojanja čoveka od robota i greška orijentacije mogu straćunati u robotskom koordinatnom sistemu, i predstavljaju ulaze u pozicioni kontroler. Sistem predstavljen u ovoj tezi u realnom slučaju sa senzora vizije dobija podatke o poziciji čoveka u koordinatnom sistemu kamere Z_{HC} , X_{HC} , i nakon transformacije koordinata na osnovu jednačine (1.7) dobijamo koordinate položaja čoveka u robotskom koordinatnom sistemu tj. x_{HR} , y_{HR} . Međutim, za testiranje performansi pozicionog kontrolera, dovoljno je bilo simulirati kretanje čoveka u globalnom koordinatnom sistemu, zatim izvršiti tranformaciju koordinata čoveka iz globalnog u robotski koorinatni sistem na osnovu jednačine (4.22).

$$d = \sqrt{e_{xR}^2 + e_{yR}^2} = \sqrt{x_{HR}^2 + y_{HR}^2} \quad (4.23)$$

$$\varphi = \arctg \frac{e_{yR}}{e_{xR}} = \arctg \frac{y_{HR}}{x_{HR}} \quad (4.24)$$

Matlab Simulink model koršćen za simulaciju pozicionog kontrolera je prikazan na Slici 4.10:



Slika 4.10. Strukturna šema za simulaciju i testiranje pozicionog kontrolera

Korišćenjem strukturne šeme sa Slike 4.10 i šeme generatora putanje čoveka, može se testirati pozicioni kontroler sa različitim referentnim putanjama i različitim inicijalnim pozicijama za robota.

U razmatranom robotskom sistemu željeno je još da robot prati čoveka na rastojanju od 1.5 m, i prema tome robot više ne treba da dotigne čoveka, već neku tačku koja se nalazi na rastojanju od 1.5 od čoveka. Stoga se greška pozicije, umanjuje za 1.5 m sračunava kao:

$$d = \sqrt{x_{HR}^2 + y_{HR}^2} - 1.5 \quad (4.25)$$

Da bi se lakše testirale performanse predložene upravljačke šeme, kontroler se testira za slučaj kada je cilj upravljanja da je rastojanje između robota i čoveka jednako nuli - jednačina (4.24). Takođe prikazano je i ponašanje kada je cilj upravljanja da je rastojanje između robota i čoveka jednako 1.5 m - jednačina (4.25). U oba slučaja zeljena greška orijentacije je jednaka nuli $\varphi = 0$. Maksimalna brzina koju može da dostigne robot je $v=1.5$ m/s.

4.3.2.1. Testiranje upravljačke šeme za konstantu brzinu čoveka i željeno rastojanje $d=0$ m

Parametri nelinearnog kontrolera iznose $K_p=1.5$ i $K_d=2.5$.

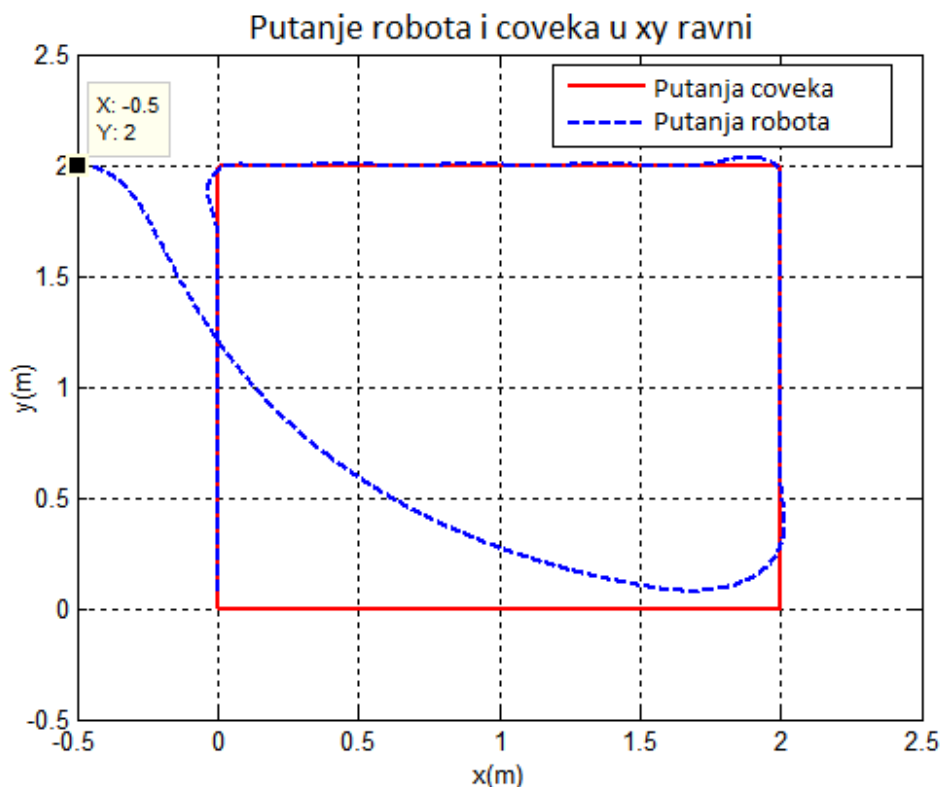
Referentna putanja čoveka : kvadrat

Jednačina putanje: Šema generatora data u Dodatku A.

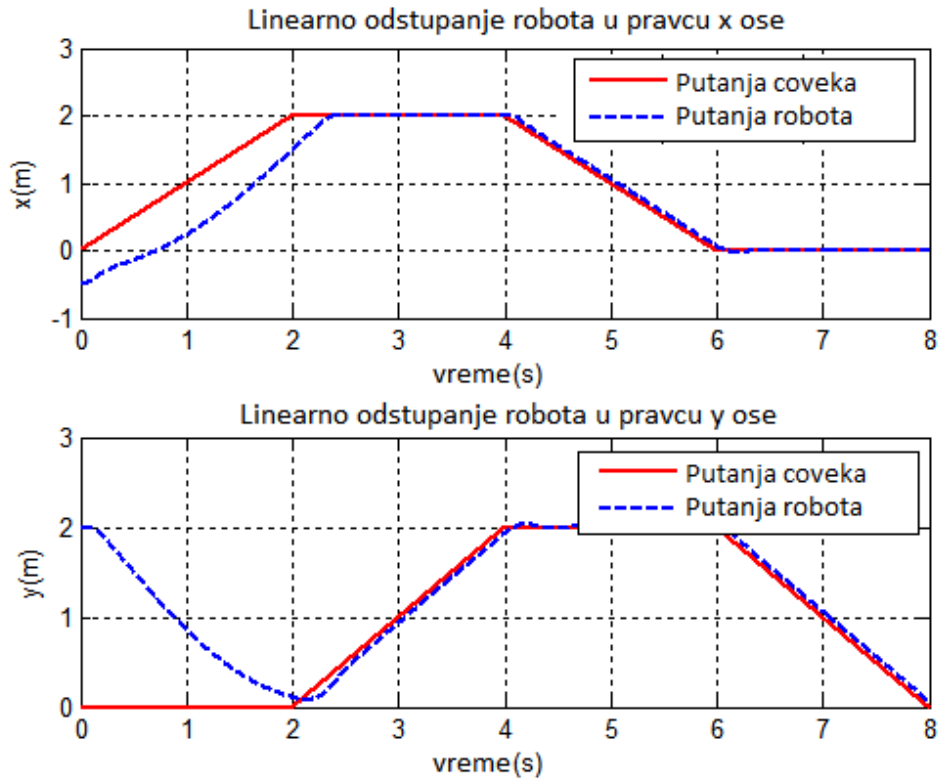
Inicijalna pozicija robota: $(-0.5, 2, 180^\circ)$.

Inicijalna pozicija čoveka: $x_t(0) = 0, y_t(0) = 0$.

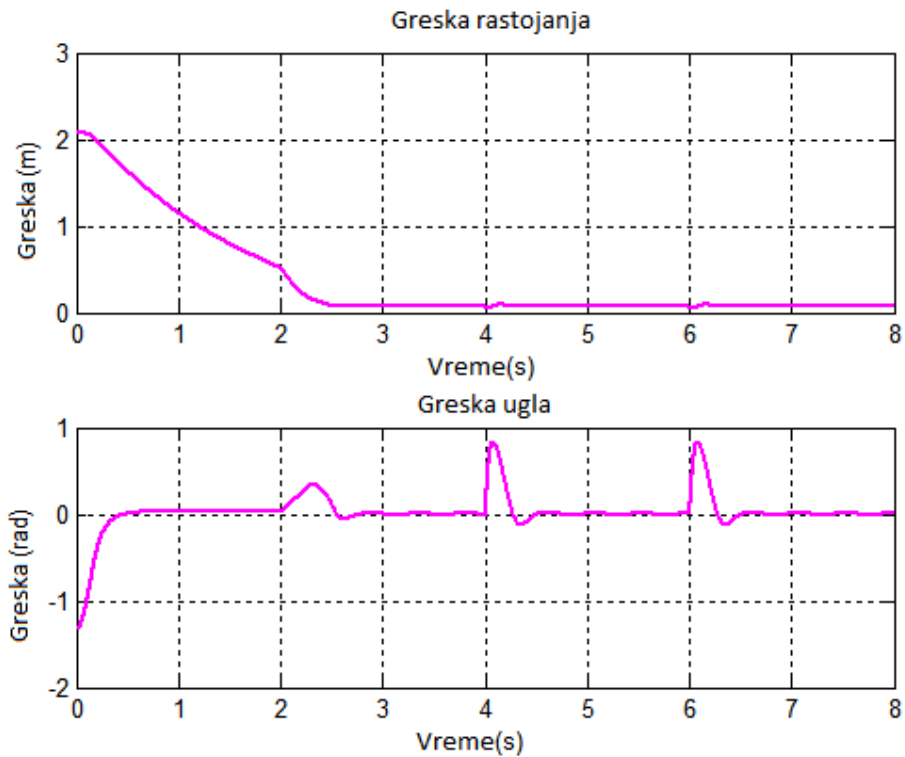
Željeno rastojanje između robota i čoveka: $d=0$ m.



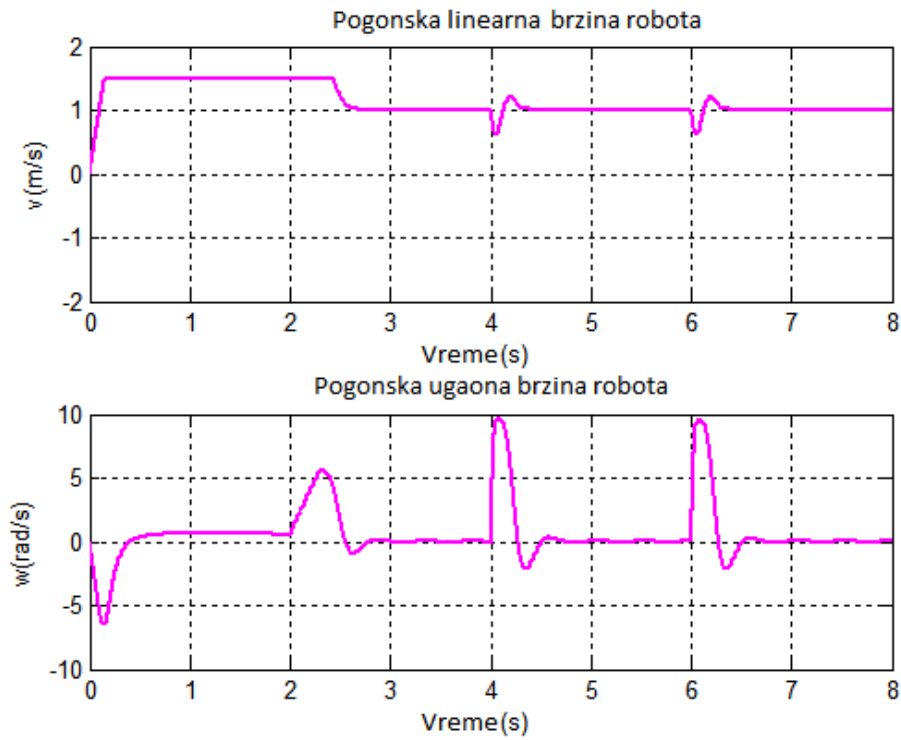
Slika 4.11. Putanje čoveka i robota-putanja u obliku kvadrata, $d=0$ m



Slika 4.12. Linerno odstupanje robota u x i y pravcu u toku vremena-putanja u obliku kvadrata, $d=0$ m



Slika 4.13. Greška praćenja željenog rastojanja i ugaona greška praćenja (ulazi u pozicioni kontroler)-putanja u obliku kvadrata, $d=0$ m



Slika 4.14. Pogonska linearna v i ugaona brzina ω robota tokom vremena-putanja u obliku kvadrata,
 $d=0$ m

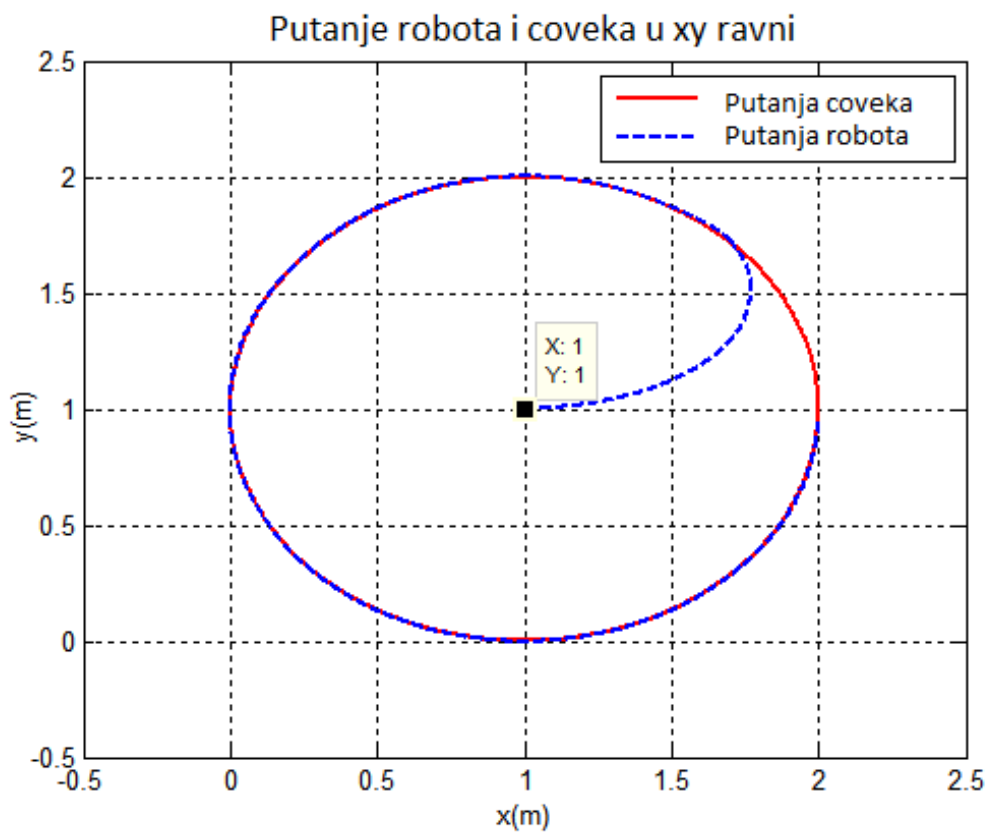
Referentna putanja čoveka : *krug*

Jednačina putanje: $x_t = 1 + \cos t, y_t = 1 + \sin t$.

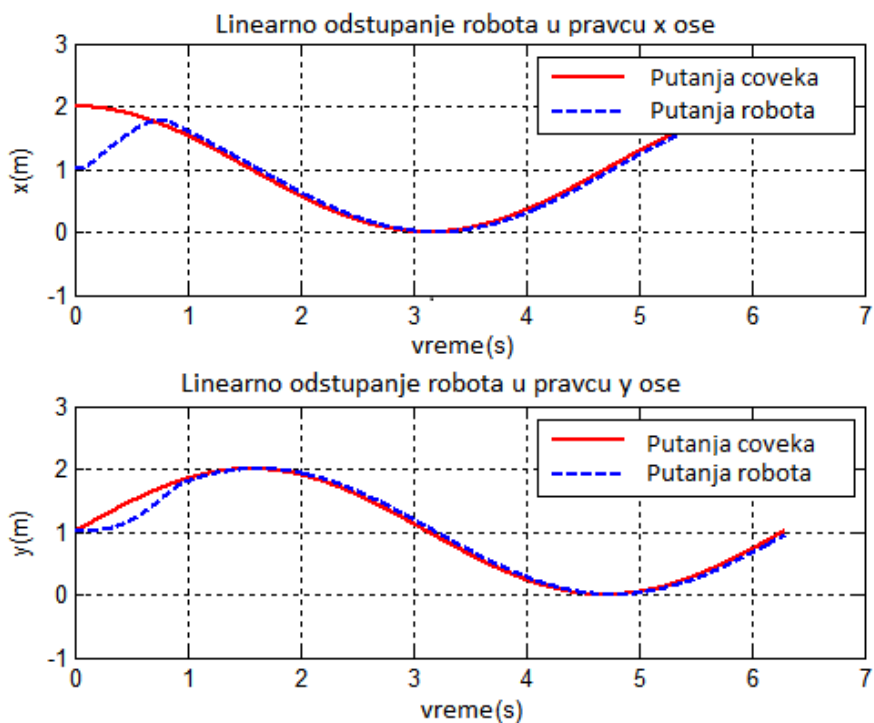
Inicijalna pozicija robota: $(1, 1, 0^0)$.

Inicijalna pozicija čoveka: $x_t(0) = 2, y_t(0) = 1$.

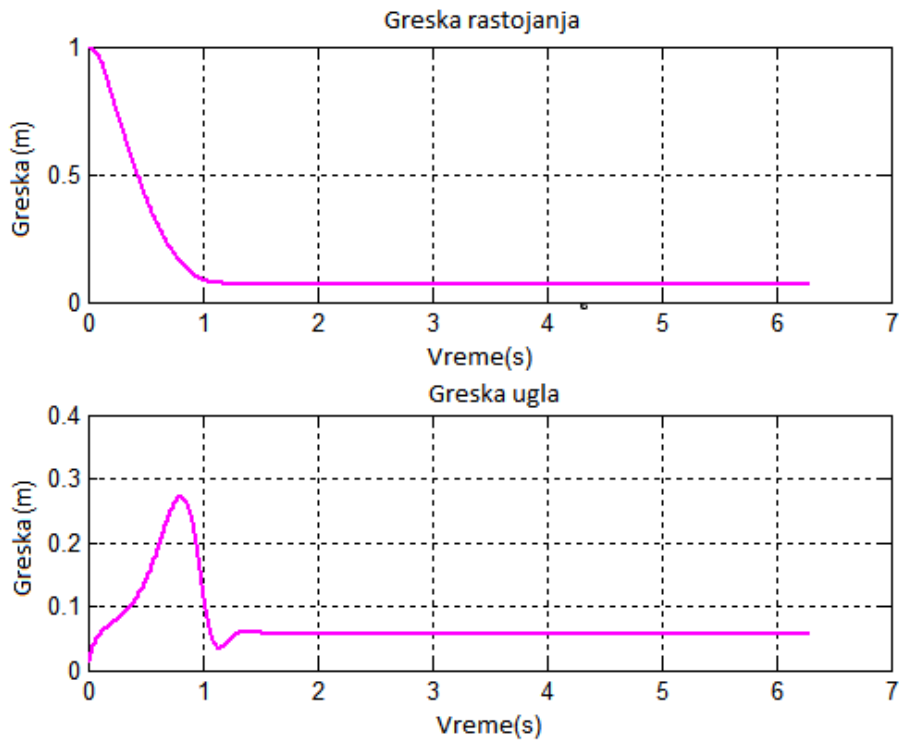
Željeno rastojanje između robota i čoveka: $d=0$ m.



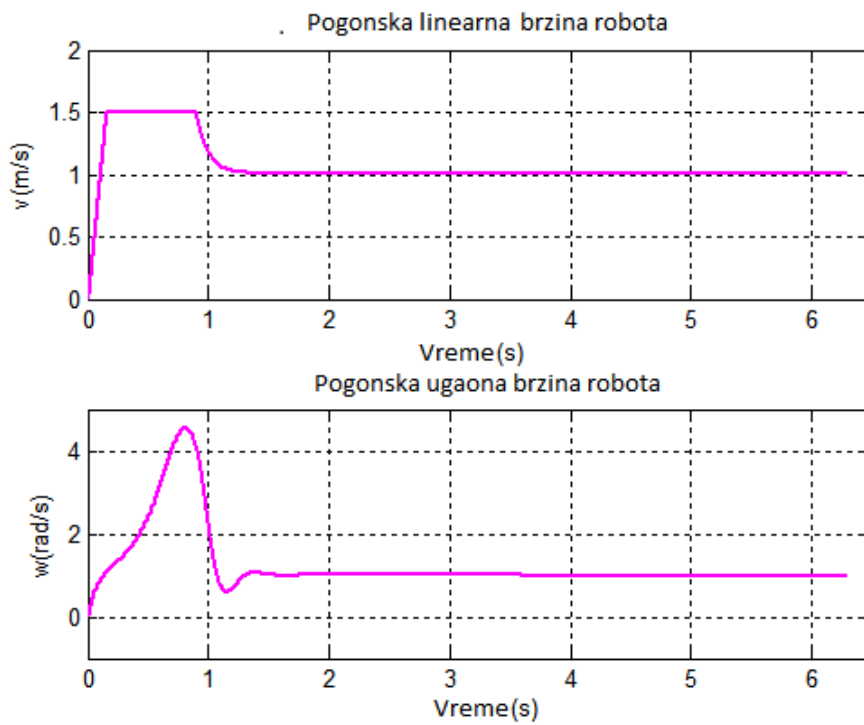
Slika 4.15. Putanje čoveka i robota-putanja u obliku kruga, $d=0$ m



Slika 4.16. Linerno odstupanje robota u x i y pravcu u toku vremena - putanja u obliku kruga, $d=0$ m



Slika 4.17. Greška praćenja željenog rastojanja i ugaona greška praćenja-putanja u obliku kruga (ulazi u pozicioni kontroler), $d=0$ m



Slika 4.18. Pogonska linearna v i ugaona brzina ω robota tokom vremena-putanja u obliku kruga, $d=0$ m

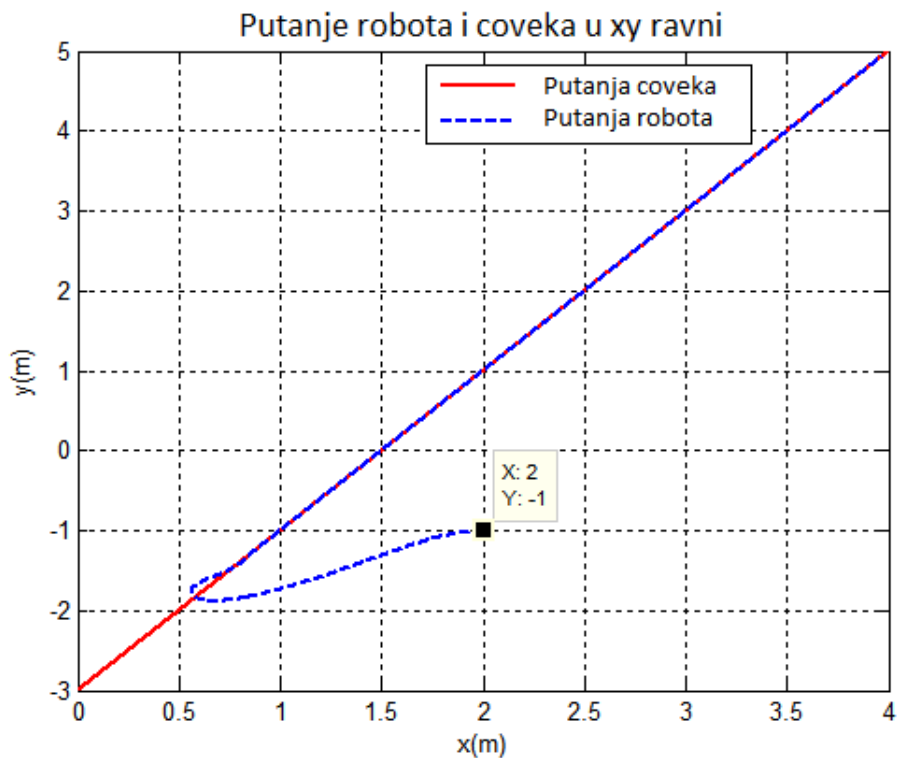
Referentna putanja čoveka : *prava*

Jednačina putanje: $x_t = 0.5 \cdot t, y_t = 2 \cdot x - 3$.

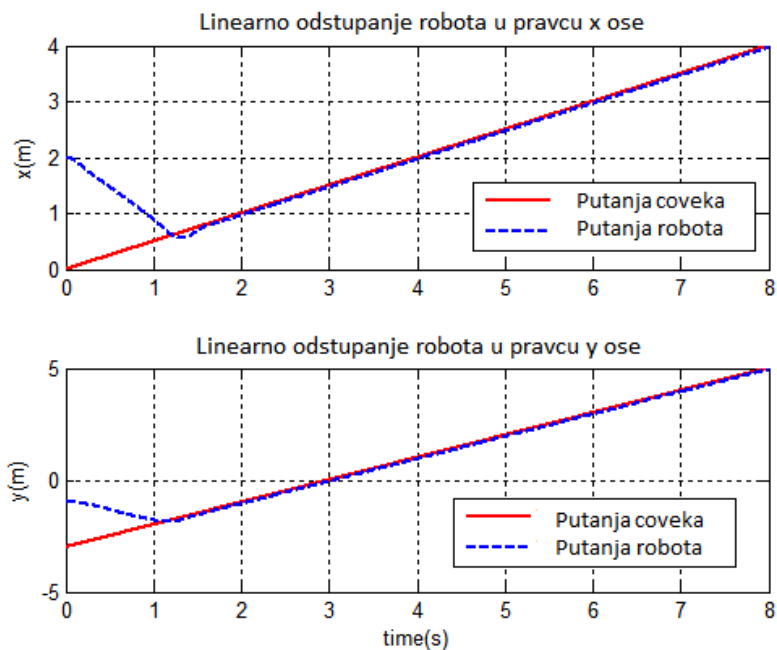
Inicijalna pozicija robota: $(2, -1, 180^\circ)$.

Inicijalna pozicija čoveka: $x_t(0) = 0, y_t(0) = -3$.

Željeno rastojanje između robota i čoveka: $d=0$ m.

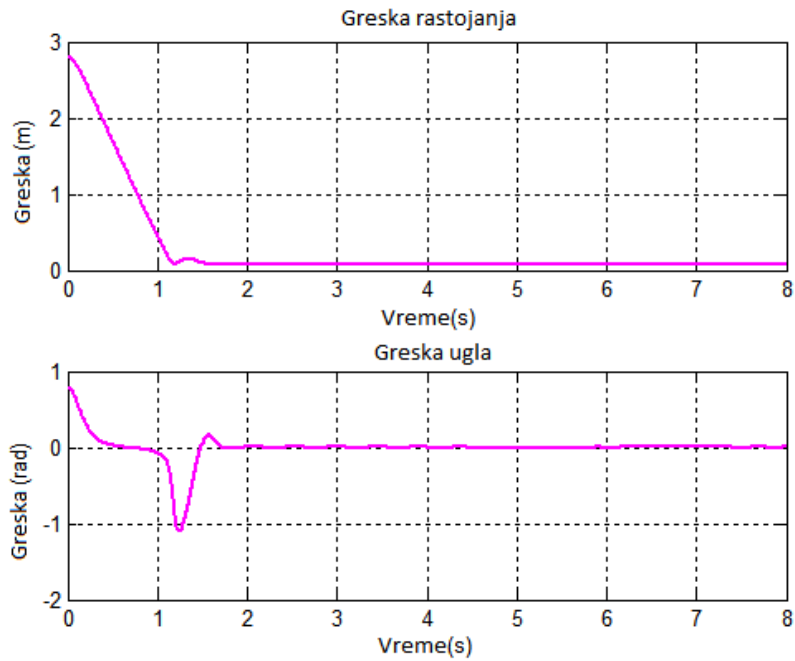


Slika 4.19. Putanje čoveka i robota-putanja u obliku prave linije, $d=0$ m

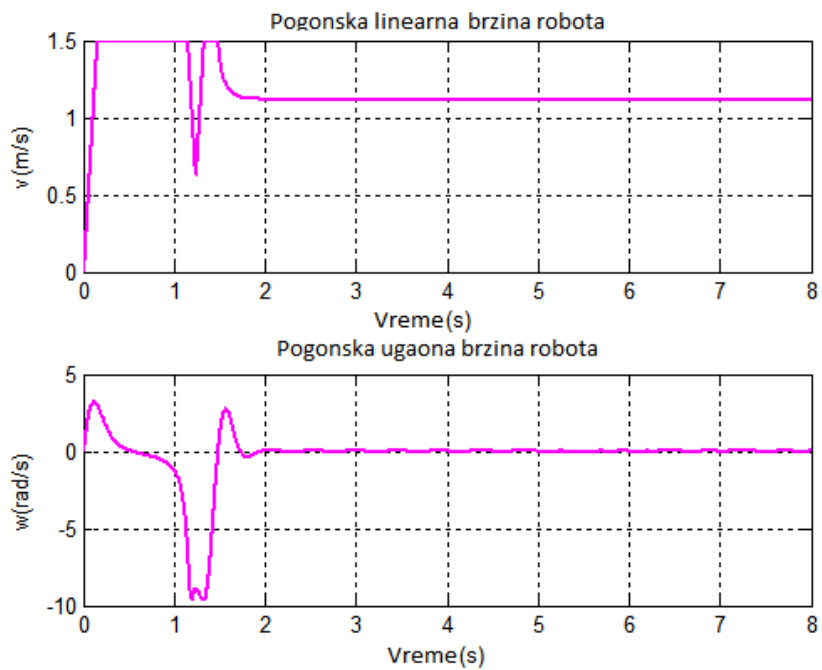


Slika 4.20. Linerno odstupanje robota u x i y pravcu u toku vremena-putanja obliku prave linije,

$d=0$



Slika 4.21. Greška praćenja željenog rastojanja i ugaona greška praćenja-putanja u obliku prave linije (ulazi u pozicioni kontroler), $d=0$ m



Slika 4.22. Pogonska linearna v i ugaona brzina ω robota tokom vremena-putanja u obliku prave linije, $d=0$ m

4.3.2.2. Testiranje upravljačke šeme za konstantu brzinu čoveka i željeno rastojanje $d=1.5m$

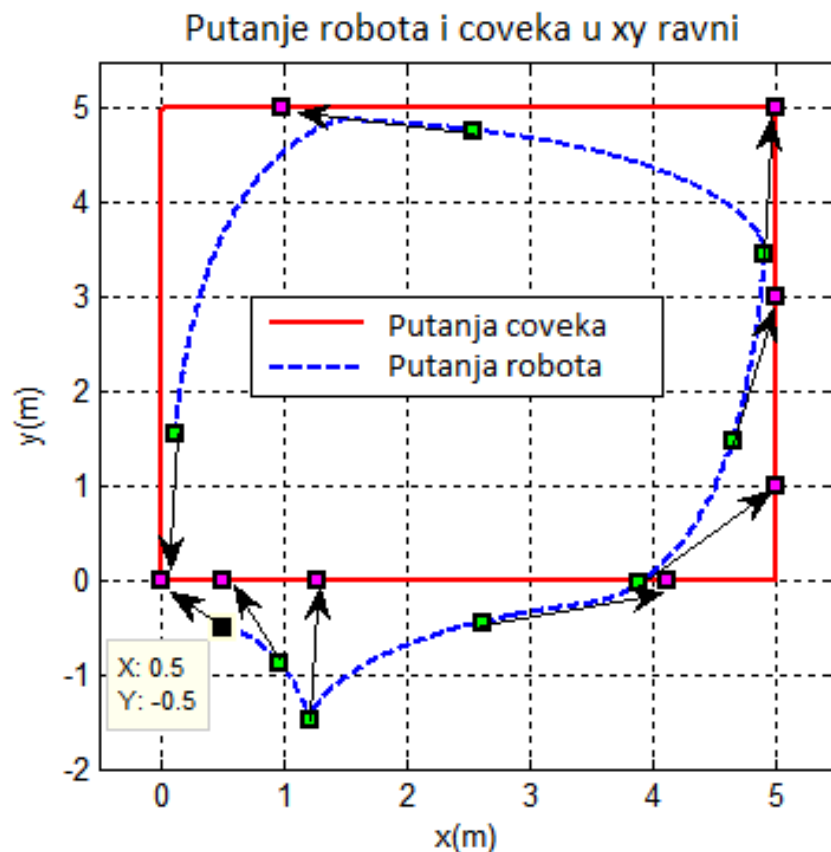
Referentna putanja čoveka : kvadrat

Jednačina putanje: Šema generatora data u Dodatku A.

Inicijalna pozicija robota: $(0.5, -0.5, 180^\circ)$.

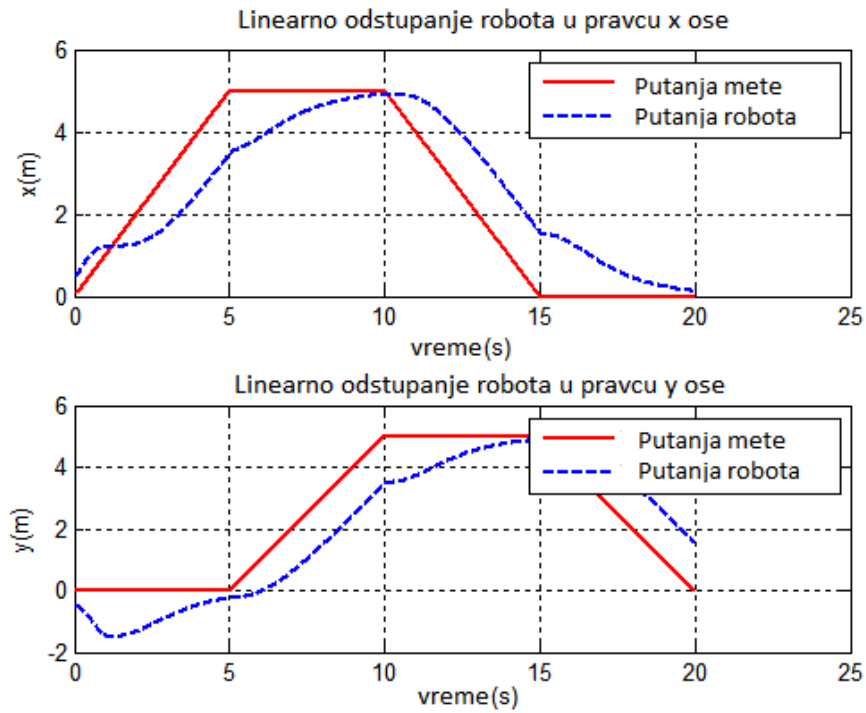
Inicijalna pozicija čoveka: $x_t(0) = 0, y_t(0) = 0$.

Željeno rastojanje između robota i čoveka: $d=1.5 m$.

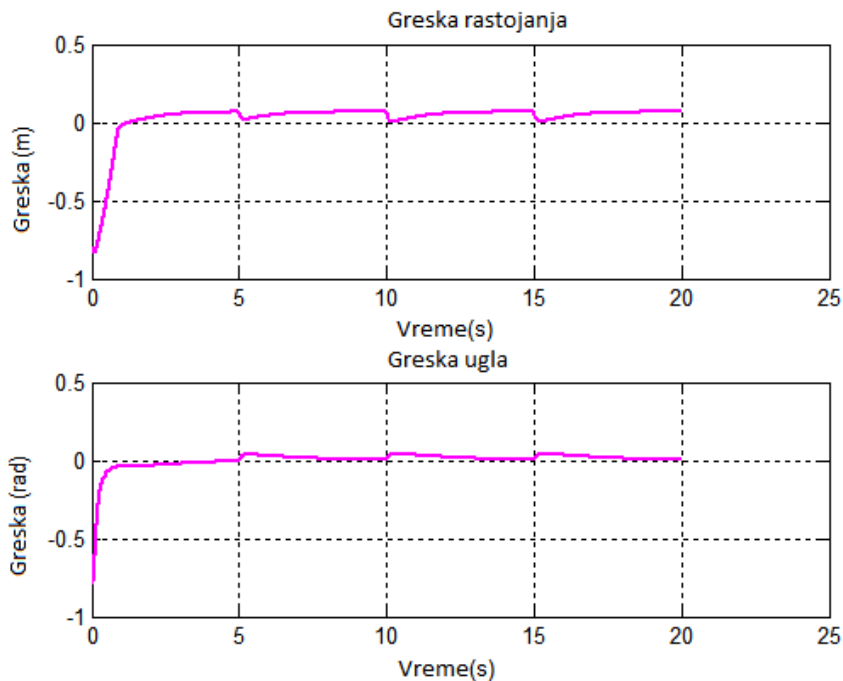


Slika 4.23. Putanje čoveka i robota -putanja u obliku kvadrata, $d=1.5 m$

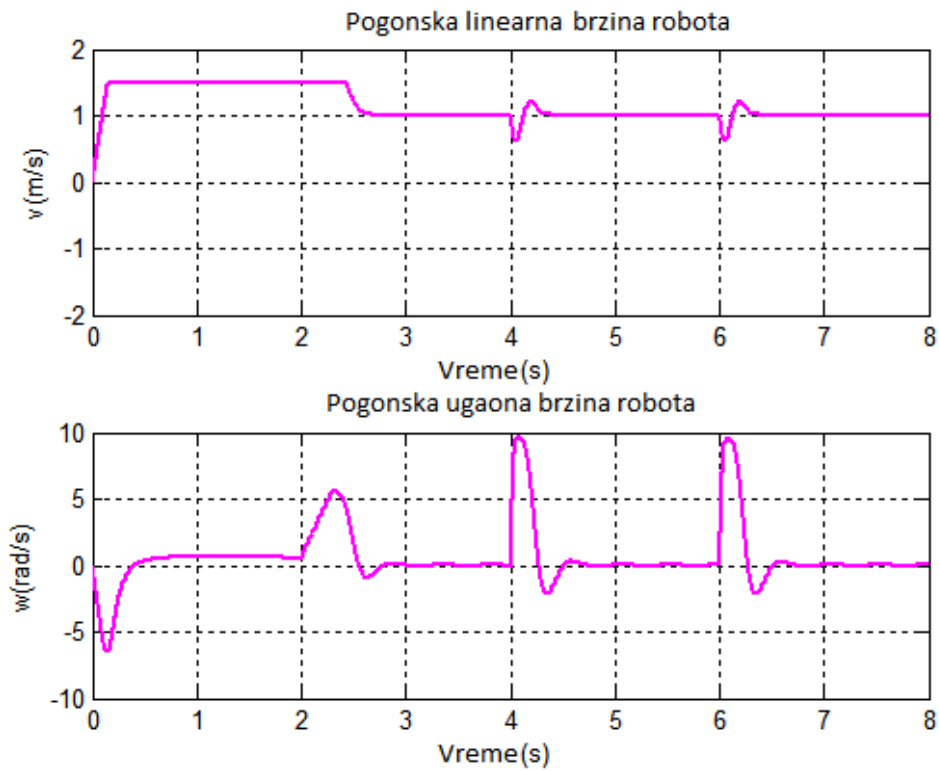
Kao što se može videti sa slike 4.23, u početnom trenutku robot se nalazi blizu čoveka. Cilj upravljanja je da robot prati čoveka održavajući konstantu udaljenost od 1.5 m. Zbog toga robot se prvo udaljava od čoveka sve dok se ne dostigne rastojanje od 1.5 m, a zatim kreće da ga prati. Na slici 4.23, su označene i povezane strelicom odgovarajuće pozicije čoveka i robota za određene trenutke u toku kretanja.



Slika 4.24. Linerno odstupanje robota u x i y pravcu u toku vremena-putanja u obliku kvadrata, $d=1.5$ m.



Slika 4.25. Greška praćenja željenog rastojanja i ugaona greška praćenja(ulazi u pozicioni kontroler)-putanja u obliku kvadrata, $d=1.5$ m



Slika 4.26. Pogonska linearna v i ugaona brzina ω robota tokom vremena-putanja u obliku kvadrata, $d=1.5$ m

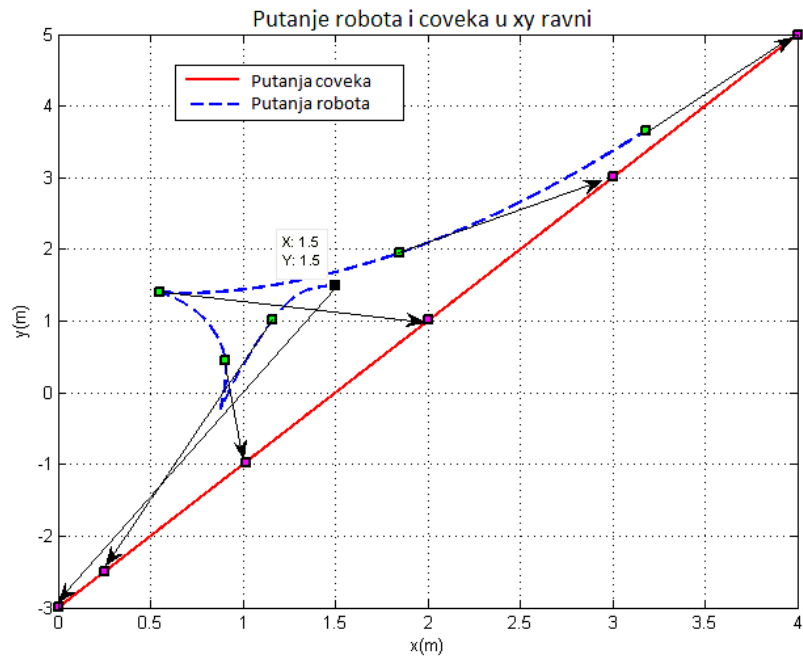
Referentna putanja čoveka : *prava*

Jednačina putanje: $x_t = 0.5 \cdot t, y_t = 2 \cdot x - 3$.

Inicijalna pozicija robota: $(1.5, 1.5, 180^0)$.

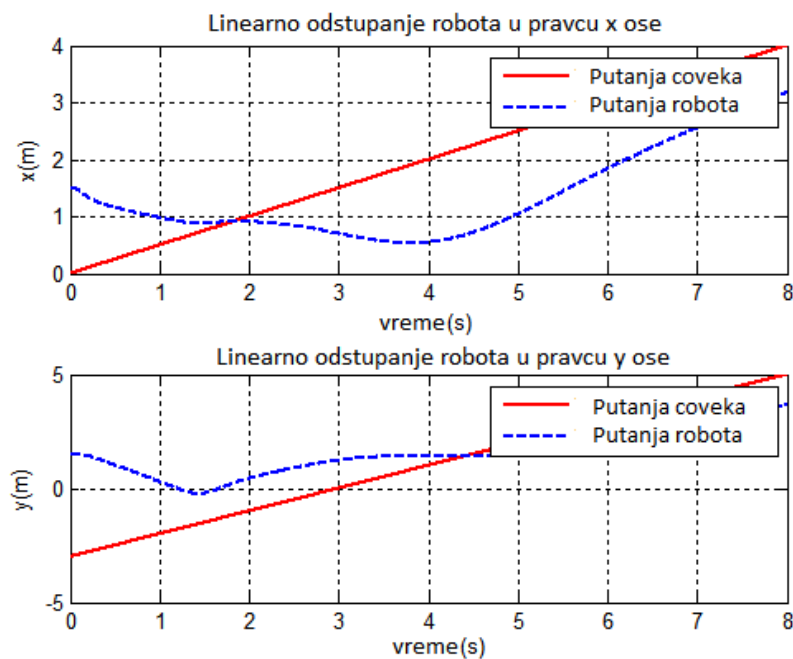
Inicijalna pozicija čoveka: $x_t(0) = 0, y_t(0) = -3$.

Željeno rastojanje između robota i čoveka: $d=1.5$ m.

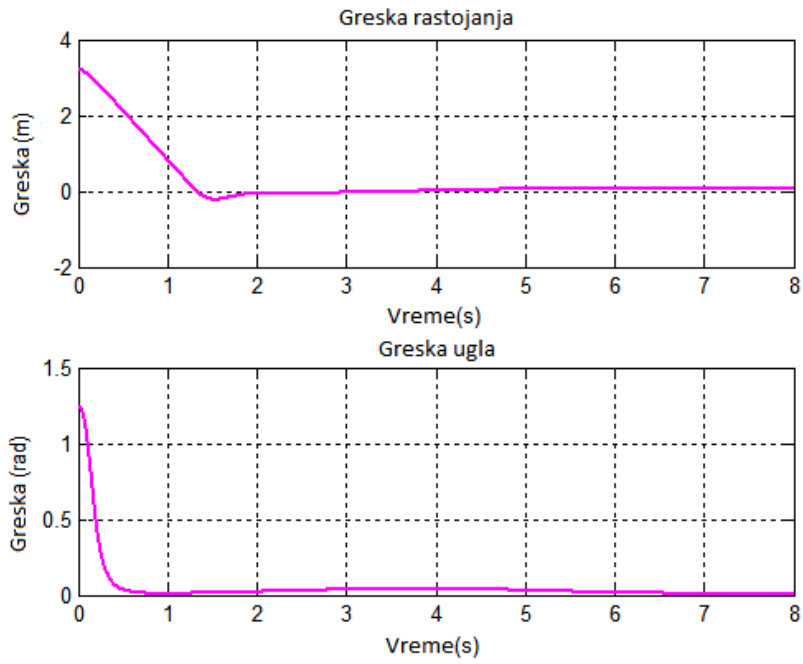


Slika 4.27. Putanja čoveka i robota-putanja u obliku prave linije, $d=1.5$ m

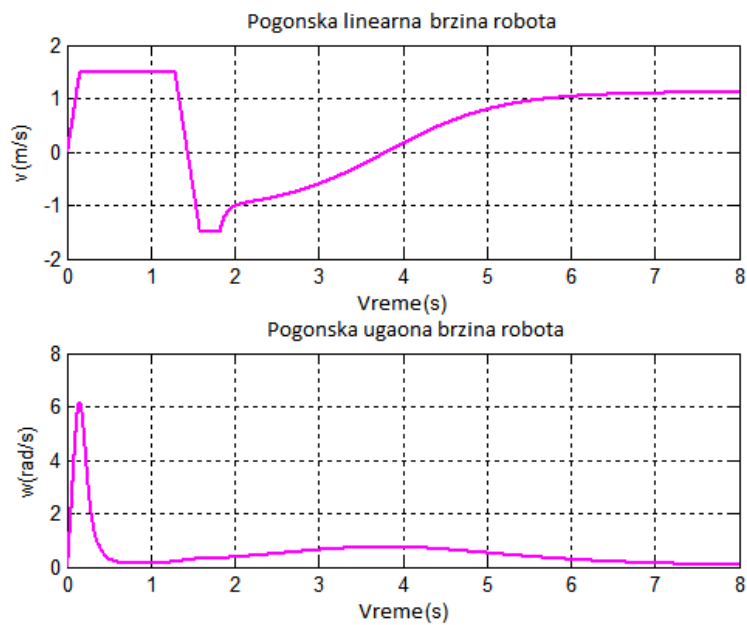
U početnom trenutku robot je udaljen od čoveka više od 1.5 m, i robot teži da mu se približi. Budući da je smer kretanja čoveka ka robota u trenutku kada je greška praćenja negativna, robot će početi da se udaljava, održavajući konstantnu udaljenost od 1.5 m. Kada greška praćenja postane ponovo pozitivna robot kreće da sledi čoveka. Na slici 4.27 su označene i povezane strelicom odgovarajuće pozicije čoveka i robota za određene trenutke u toku kretanja.



Slika 4.28. Linerno odstupanje robota u x i y pravcu u toku vremena-putanja obliku prave linije, $d=1.5$



Slika 4.29. Greška praćenja željenog rastojanja i ugaona greška praćenja-putanja u obliku prave linije (ulazi u pozicioni kontroler), $d=1.5$ m



Slika 4.30. Pogonska linearna v i ugaona brzina ω robota tokom vremena-putanja u obliku prave linije, $d=1.5$ m

4.3.2.3. Testiranje upravljačke šeme za promenljivu brzinu čoveka i željeno rastojanje $d=0$

Takođe je testiran kontroler i za slučaj kada je brzina kretanja čoveka promenljiva.

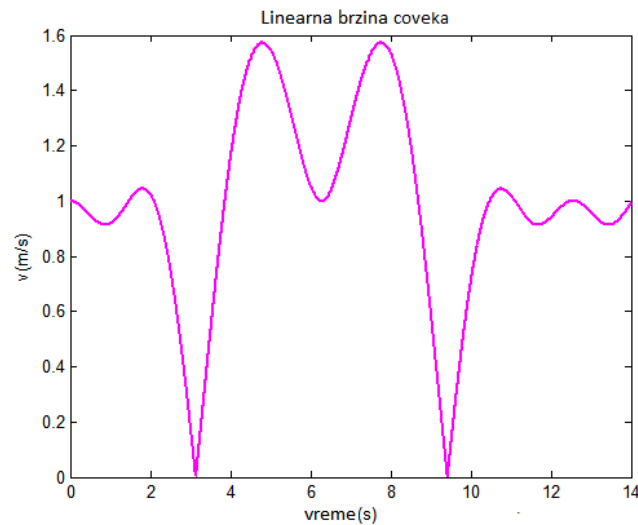
Referentna putanja čoveka: putanja oblika ∞ (simbol za beskonačno)

Jednačina putanje: $x_t = 2 \cdot \cos(0.5 \cdot t)$; $y_t = \sin(0.5 \cdot t) \cdot x$;

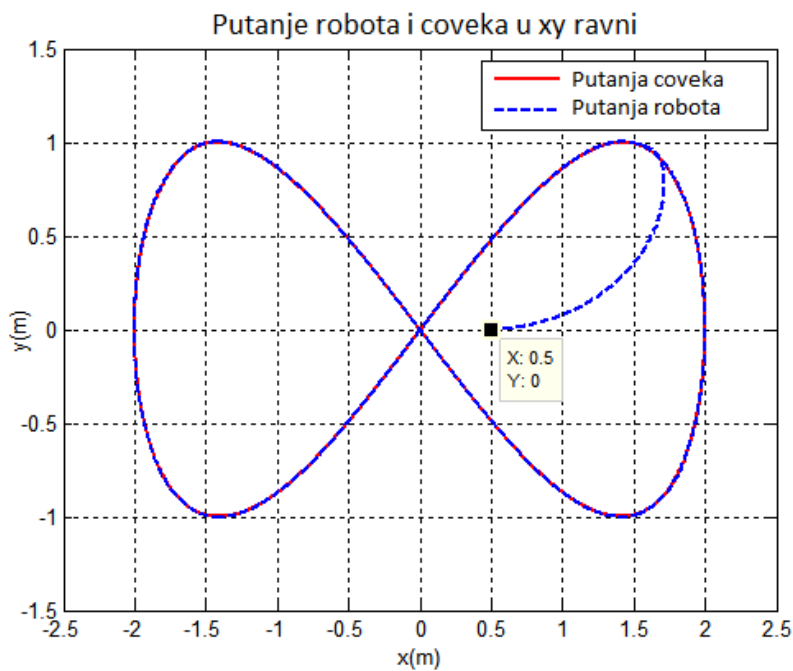
Inicijalna pozicija robota: $(0.5, 0, 0^\circ)$.

Inicijalna pozicija čoveka: $x_t(0) = 2, y_t(0) = 0$.

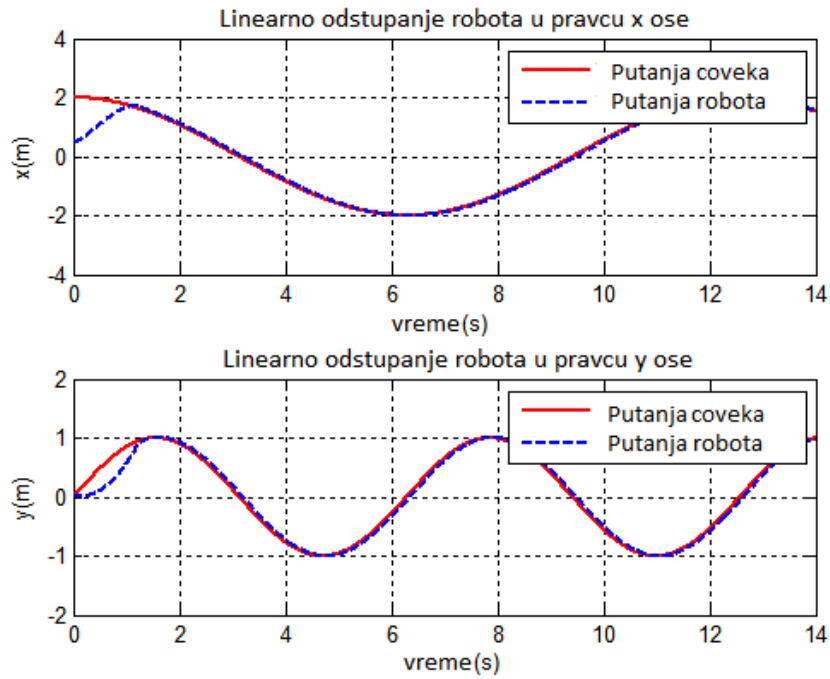
Željeno rastojanje između robota i čoveka: $d=0$ m.



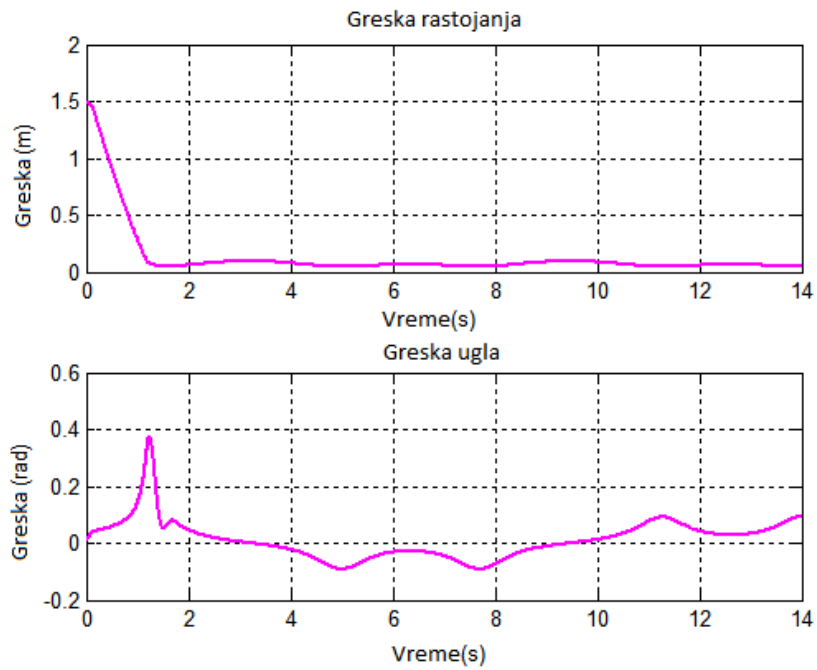
Slika 4.31. Brzina kretanja čoveka (putanja oblika ∞)



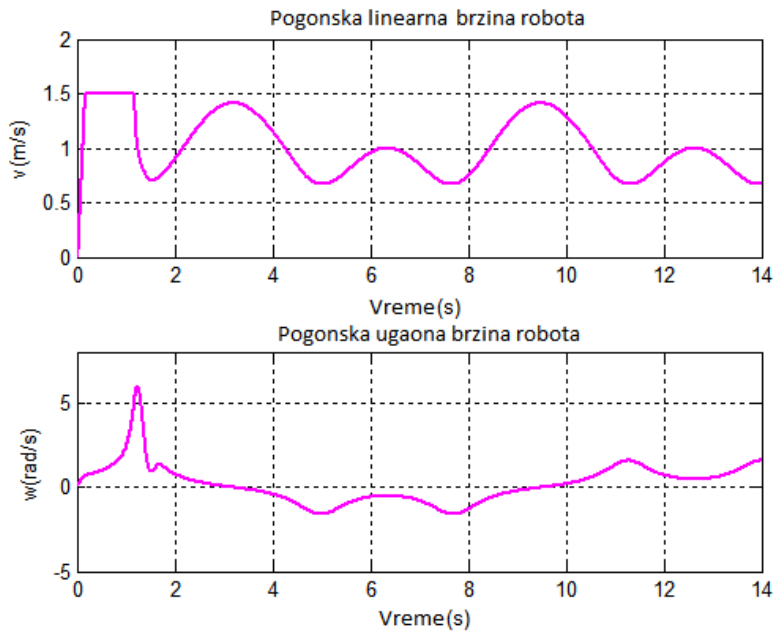
Slika 4.32. Putanje čoveka i robota-putanja u obliku ∞ , $d=0$ m



Slika 4.33. Linerno odstupanje robota u x i y pravcu u toku vremena-putanja u obliku ∞ , $d=0$ m.



Slika 4.34. Greška praćenja željenog rastojanja i ugaona greška praćenja (ulazi u pozicioni kontroler) - putanja u obliku ∞ , $d=0$ m.



Slika 4.35. Pogonska linearna v i ugaona brzina ω robota tokom vremena-putanja putanja u obliku ∞ , $d=0$ m.

Na osnovu prikazanih rezultata može se zaključiti da je greška praćenja približna nuli (svega pola centimetra) i da će robot pratiti čoveka bez obzira na oblik referentne putanje i početne pozicije robota u odnosu na referentnu putanju. Uzimajući u obzir i same dimenzije robota, može se zaključiti da predložena upravljačka šema može da omogući uspešno praćenje čoveka od strane robota.

4.4. OPTIMIZACIJA PARAMETARA MULTIVARIJABILNOG PD KONTROLERA ROJEVIMA ČESTICA

Optimizacija rojem čestica (Particle Swarm Optimisation, PSO) je stohastički algoritam koji se bazira na populaciji rešenja i jedan je od najmlađih u kategoriji meta-heurističkih optimizacija. Predstavljen je prvi put 1995. godine od strane Kennedy-ja i Eberhart-a [64]. Algoritam je inspirisan socijalnim ponašanjem jata ptica koje lete u potrazi za hranom. Osnovu algoritma čini populacija koja se sastoji od niza jedinki (n čestica) koje lete kroz višedimenzionalni prostor koji pretražuju i pri tome svoj položaj menjaju na osnovu vlastitog iskustva i iskustva ostalih čestica u roju (čime se modeliraju socijalne interakcije između jedinki). Prilikom određivanja smera kretanja, svaka jedinka u određenoj meri uzima u obzir svoje, do tada, pronađeno najbolje rešenje (individualni faktor) i najbolje rešenje ostatka roja (socijalni faktor).

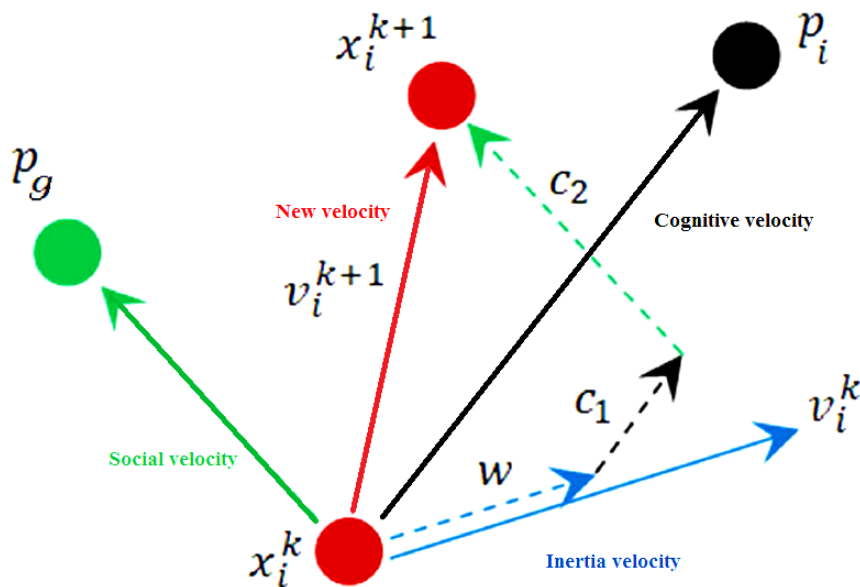
Poticija svake i -te čestice u roju predstavljena je D -dimenzionalnim vektorom $x_i = (x_1, x_2, \dots, x_D)$. Brzina kretanja (promena pozicije po inetraciji) čestice može se takođe predstaviti D-

dimenzionalnim vektorom $\mathbf{v}_i = (v_1, v_2, \dots, v_D)$. Svaka čestica unutar roja pamti koordinate unutar prostora problema koje predstavljaju najbolje dosad postignuto rešenje te čestice (*personal best*) i označićemo ga sa $\mathbf{p}_i = (p_1, p_2, \dots, p_D)$. Toplogija korišćena u ovom radu je takva da se za sve čestice predpostavlja da su susedi i \mathbf{g} je indeks čestice u roju koja je na najboljoj poziciji, i tada \mathbf{p}_g (*global best*) postaje najbolje pronađeno rešenje do tada, a brzina i čestica i njihove nove pozicije se ažuriraju prema sledećima dvema jednačinama:

$$\mathbf{v}_i^{t+1} = \omega \mathbf{v}_i^t + c_1 r_1 (\mathbf{p}_i^t - \mathbf{x}_i^t) + c_2 r_2 (\mathbf{p}_g^t - \mathbf{x}_i^t) \quad (4.26)$$

$$\mathbf{x}_i^{t+1} = \mathbf{x}_i^t + \mathbf{v}_i^{t+1}, \quad (4.27)$$

gde r_1 i r_2 predstavljaju dva nasumična vektora čije komponente poprimaju vrijednosti iz intervala $[0,1]$, c_1 je kognitivni parametar koji reguliše relativnu brzinu čestice prema svom ličnom najboljem rešenju, dok je c_2 društveni parametar koji predstavlja važnost društva, odnosno globalno najprikladnijeg rešenja i reguliše relativnu brzinu čestice prema globalno najboljem rešenju [65][66][67].



Slika 4.36. Prikaz ažuriranja brzine i pozicije čestica kod optimizacije rojevima čestica

Faktor inercije w je parametar koji kontroliše dinamiku letenja. Koristi se za uravnoteženje globalne i lokalne sposobnosti pretrage. Velika vrednost faktora inercije je pogodnija za globalno istraživanje novih područja, dok su niže vrednosti pogodnije za lokalno pretraživanje. Postoji nekoliko različitih varijanti faktora inercije koji se koriste za poboljšanje performansi optimizacije

rojem čestica. Pregled 15 najpopularnijih strategija za izbor faktora inercije i upoređivanje njihovih performansi na nekoliko problema optimizacije je dat u radu [68].

Za nalaženje optimalnih parametara multivarijabilnog PD kontrolera, u cilju uspešnog praćenja zadatih upravljačkih brzina, u ovoj tezi, korišćen je linearno opadajući faktor inercije:

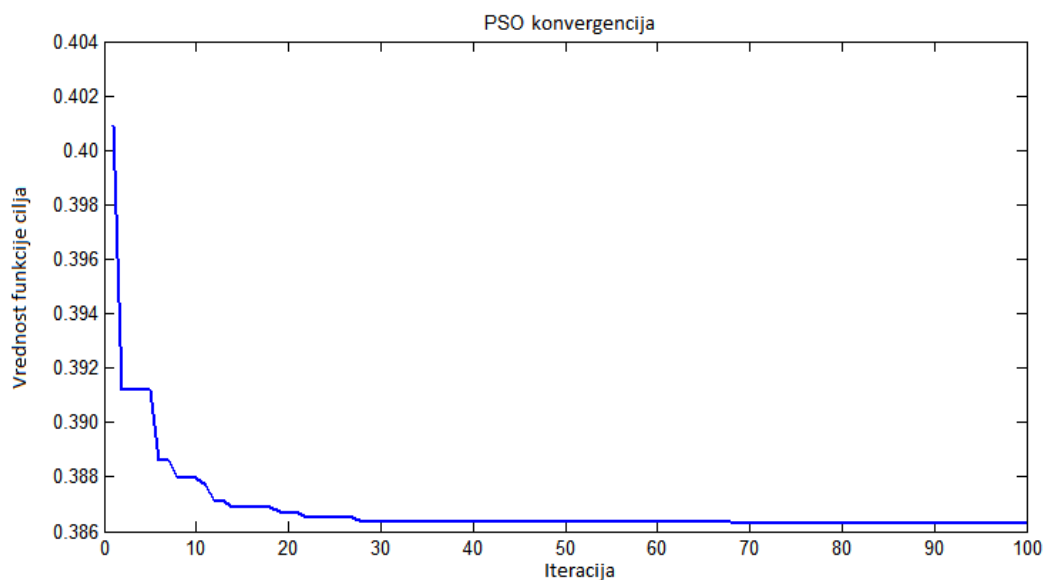
$$w_i = w_{max} - \left(\frac{w_{max} - w_{min}}{i_{max}} \right) \quad (4.28)$$

w_i i w_{max} predstavljaju inicijalnu i krajnju vrednost faktora inercije, dok je i_{max} maksimalan broj iteracija.

Funkcija cilja je definisana kao kriterijum konvergencije da bi se došlo do optimalnog rešenja. Izabrana fitness funkcija je funkcija linearne i ugaone greške brzine i data je sledećom jednačinom:

$$F_{obj} = \sum_i^N (e_v^2(i) + e_w^2(i)) \quad (4.29)$$

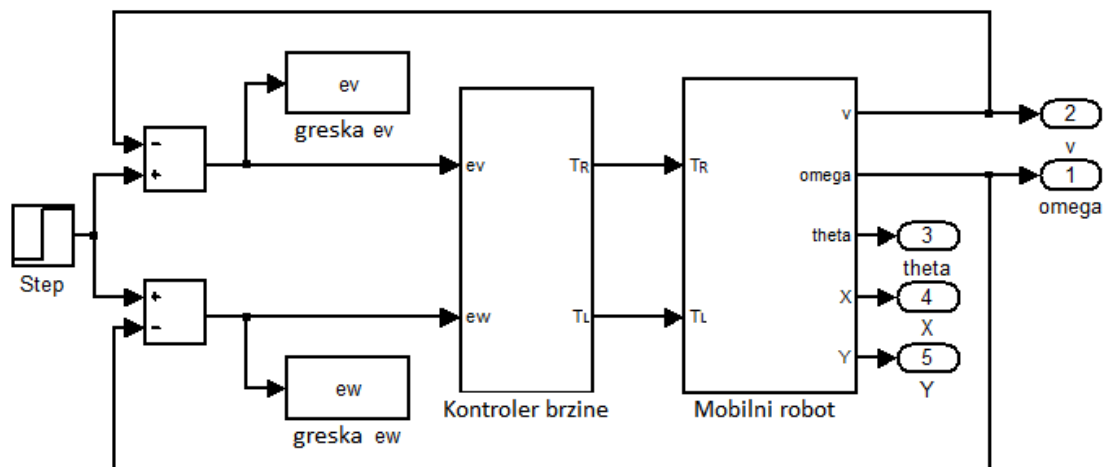
Proces nalaženja najbolje vrednosti fitness funkcije kroz iteracije prikazan je na slici 4.37.



Slika 4.37. Konvergencija fitness funkcije za kontroler brzine sa brojem iteracija

U svakoj generaciji je $n=30$ čestica (*eng.* particles). Vrednosti parametra PSO (jednačina 4.26) koji se koriste u optimizaciji su: $c_1=c_2=2.1$, $w_{max}=0.9$, $w_{min}=0.3$.

Na slici 4.38. je prikazan blok dijagram korišćen za dobijanje potrebnih podataka za obavljanje optimizacije.



Slika 4.38. Dijagram za dobijanje podataka za obavljanje optimizacije

Ovaj algoritam dobija sve veću popularnost među istraživačima kao efikasna tehnika za rešavanje složenih problema, a pritom je veoma lak za implementaciju, jer potrebno je podesiti svega nekoliko parametara.

5. STOHAŠTIČKI ALGORITMI ZA REŠAVANJE PROBLEMA PRAĆENJA LJUDI NA OSNOVU PODATAKA DOBIJENIM SENZOROM ROBOTSKE VIZIJE

Budući da je slobodno kretanje čoveka nepredvidljivo, nemoguće je unapred znati tačnu putanju kretanja zbog čega se koristi stohastički model za opisivanje kretanja čoveka. Postoje nekoliko modela kretanja kojim se opisuje kretanje pokretnog objekta, tj. čoveka: Braunovo kretanje (*eng.* Brownian motion), model sa konstantnom brzinom (*eng.* Constant velocity), model sa konstantnim ubrzanjem (*eng.* Constant acceleration) i drugi modeli višeg reda [109].

Najjednostavniji model je Braunov model kretanja koji opisuje stanja kao što su lokacija objekta, oblik i td. Braunov model je generički i pogodan je za širok spektar trajektorija, čineći ga korisnim za situacije gde ne postoji jasan obrazac prilikom kretanja.

Za opisivanje kretanja čoveka mnogo je pogodniji model sa konstantnom brzinom. Ovaj model procenjuje poziciju i brzinu pokretnog objekta-čoveka pri čemu ubrzanje predstavlja poremećaj tj. šum procesa. Ovaj model kretanja poseduje dodatnu glatkost u odnosu na Braunov model. Takođe, u praksi se koristi i model konstantnog ubrzanja gde se pored pozicije i brzine procenjuje i ubrzanje, dok semoment modeluje kao šum procesa. Modeli višeg reda se najčešće ne koriste kod praćenja čoveka robotskim sistemima.

Kao što je opisano u poglavlju 4, za potrebe upravljačkog bloka neophodno je sračunati rastojanje d (rastojanje između čoveka i robota) i ugao orijentacije φ u odnosu na osu X_R u robotskom koordinatnom sistemu. Ovi parametri se dobijaju na osnovu koordinata centra mase čoveka x_{HC} i z_{HC} estimiranih u koordinatnom sistemu kamere koje se prvo transformišu na osnovu jednačine (2.7) u položaj čoveka u koordinatnom sistemu robota. Tako transformisane koordinate se onda vode u upravljački blok.

Kao što je već rečeno, na osnovu jednačine (1.7), ose X_R i Z_c , Y_R i X_c su paralelene i translacije T_x i T_y su jednake nuli.

Prema tome rastojanje čoveka od robota d i orijentacija φ jednaka su:

$$d = \sqrt{x_{HC}^2 + z_{HC}^2} = \sqrt{x_{HR}^2 + y_{HR}^2} \quad (5.1)$$

$$\varphi = \arctg \frac{x_{HC}}{z_{HC}} = \arctg \frac{y_{HR}}{x_{HR}} \quad (5.2)$$

gde su x_{HC} i z_{HC} koordinate čoveka u horizontalnoj ravni u odnosu na koordinatni sistem kamere, a y_{HR} i x_{HR} koordinate čoveka u horizontalnoj ravni u odnosu na koordinatni sistem robota. U ovoj

tezi razmatra se model kretanja čoveka sa konstantnom brzinom i na osnovu kinematskih jednačina pređeni put S_k jednak je:

$$\begin{aligned} S_k &= S_{k-1} + V_{k-1} \cdot \Delta t \\ V_k &= V_{k-1} \end{aligned} \quad (5.3)$$

gde vektor S_k predstavlja poziciju čoveka u trenutku vremena k , u odnosu na koordinatni sistem kamere, vektor V_k predstavlja brzinu u xi zpravcima, dok Δt predstavlja interval odabiranja.

Prema tome vektor stanja u prikazanom sistemu za praćenje čoveka jednak je:

$$\mathbf{x}_k = [S_k \quad V_k]^T = [x_{HC} \quad z_{HC} \quad \dot{x}_{HC} \quad \dot{z}_{HC}]^T \quad (5.4)$$

Drugi član uključen u proces estimacije pozicije i brzine čoveka je model merenja. Taj član opisuje proces merenja i zavisi od vrste senzora koji se koristi, a koji je u razmatranom sistemu kamera.

Kao što je već rečeno u poglavlju 1, modul detekcije čoveka je sposoban da detektuje čoveka u video slikama kamere i da izračuna 3D koordinate čovekovog centra mase u koordinatnom sistemu kamere x_{HC} , y_{HC} , z_{HC} . Međutim, za potrebe ove teze koristi se samo pozicija čoveka u horizontalnoj 2D ravni i na osnovu tih podataka određuje se udaljenost čoveka od robota. Modul detekcije čoveka je takođe sposoban da konstantno dostavlja te podatke u cilju uspešnog praćenja i održavanja tačno određene udaljenosti između čoveka i robota.

Na osnovu navedenog, vektor merenja je:

$$\mathbf{y}_k = [S_k]^T = [x_{HC} \quad z_{HC}]^T \quad (5.5)$$

Na osnovu kinematskih jednačina modela, vektora stanja i vektora merenja može se zaključiti da je razmatrani problem linearan.

Budući da modul praćenja treba da bude primenljiv kod različitih robotskih platformi i kod različitih senzora detekcije položaja čoveka, bilo je potrebno razviti algoritme za slučajeve kada se kao rezultati merenja dobijaju rastojanje d i ugao φ .

Vektor merenja, u ovom slučaju jednak je:

$$\mathbf{y}_k = [d \quad \varphi]^T \quad (5.6)$$

Na osnovu jednačina (5.1), (5.2) i (5.6) sledi da je jednačina merenja nelinearna:

$$\mathbf{y}_k = h(\mathbf{x}_k + \mathbf{v}_k) \quad (5.7)$$

Usled poremećaja kojima je izložen sistem kao što su promena osvetljenja, promena ugla snimanja i prisustvo mnogo različitih objekata u sceni, vizija često ne daje pouzdane podatke o poziciji čoveka ili dolazi do potpunog izostanka tih podataka. U slučaju kada vizija ne šalje podatke (sistem za detekciju sa stereo vizijom daje vrednost NaN (*eng.* Not a Number), dok su kod sistema sa 3D senzorom sve vrednosti jednake nuli, što znači da sistem više ne detektuje čoveka), kod standardnih algoritama za Kalmanov, prošireni Kalmanov i Partikle filter, može doći do greške prilikom estimacije ili pak filter neće uopšte raditi.

U cilju prevazilaženja ovog problema, u ovoj tezi, integrisani su estimatori kod kojih se vrši ispitivanje da li modul detekcije daje podatke, drugim rečima u predstavljenom sistemu integrisan je dodatni korak gde se ispituje da li nam detekcija šalje podatke. Ukoliko modul detekcije ne šalje podatke, modul praćenja će nastaviti da estimira moguću poziciju čoveka na osnovu filtera za estimaciju koji su opisani dalje u ovom poglavlju.

5.1. BAJESOVI FILTERI – TEORIJSKA OSNOVA

Bajesovi filteri probalistički procenjuju stanje dinamičkog sistema na osnovu opažanja koja su pritom zagađena šumom. Kod procene lokacije, stanje je lokacija osobe ili objekta i senzora. U slučaju razmatranog sistema, kamere obezbeđuje opservaciju o stanju. Stanje može biti 2D pozicija ili kompleksan vektor koji uključuje 3D poziciju i orijentaciju čoveka [87].

Prema Bajesovoj teoriji, sve nepoznato se smatra stohastičkom promenljivo, a početna ili pretpostavljena raspodela bi trebala da bude poznata. Koristeći opservacije, estimacija može kasnije da se koriguje na osnovu izračunate gustine verovatnoće. [88]

Nelinearni dinamički sistem može se definisati sledećom jednačinom:

$$\mathbf{x}_k = f_k(\mathbf{x}_{k-1}, \mathbf{w}_k) \quad (5.8)$$

dok se stohastički opservacioni (mereni) proces opisuje sledećom jednačinom:

$$\mathbf{y}_k = h_k(\mathbf{x}_k, \mathbf{v}_k) \quad (5.9)$$

\mathbf{x}_k je vektor stanja sistema i obično je skriven ili nije opservabilan, \mathbf{w}_k je vektor šuma dinamičkog sistema, \mathbf{y}_k je opservacioni vektor, a \mathbf{v}_k je vektor šuma merenja. Determinističke funkcije f_k i h_k , povezuju prethodno stanje sa trenutnim i trenutno stanje sa opservacionim vektorom, respektivno.

U Bajesovom kontekstu, cilj je da se izračuna *posteriorna* funkcija gustine raspodele verovatnoće, (*eng.* probalistic density function - PDF), $p(\mathbf{x}_k|\mathbf{y}_{1:k})$, gde je opservacija definisana kao $\mathbf{y}_{1:k} \triangleq \{y_1, y_2, \dots, y_k\}$, [89]. Ovaraspodela semože izračunati rekurzivno u dva koraka.

U predikcionom koraku, izračunava se *a priori* funkcijagustine raspodele verovatnoća, $p(\mathbf{x}_k|\mathbf{y}_{1:k-1})$ koja je definisana Bajesovim pravilom na sledeći način:

$$p(\mathbf{x}_k|\mathbf{y}_{1:k-1}) = \int p(\mathbf{x}_k|\mathbf{x}_{k-1}, \mathbf{y}_{1:k-1})p(\mathbf{x}_{k-1}|\mathbf{y}_{1:k-1}) dx_{k-1} \quad (5.10)$$

gde je prethodna *posteriorna* funkcija gustine raspodele verovatnoća označena kao $p(\mathbf{x}_{k-1}|\mathbf{y}_{1:k-1})$ i za koju se prepostavlja da je poznata usled rekuzije, a $p(\mathbf{x}_k|\mathbf{x}_{k-1}, \mathbf{y}_{1:k-1})$ se dobija na osnovu jednačine (5.1)

Korekcionni korak generiše *posteriornu* funkciju gustine raspodele verovatnoća kao:

$$p(\mathbf{x}_k|\mathbf{y}_{1:k}) = \frac{p(\mathbf{y}_k|\mathbf{x}_k, \mathbf{y}_{1:k-1})p(\mathbf{x}_k|\mathbf{y}_{1:k-1})}{p(\mathbf{y}_k|\mathbf{y}_{1:k-1})} \quad (5.11)$$

gde je $p(\mathbf{y}_k|\mathbf{y}_{1:k-1})$ normalizaciona konstanta koja osigurava da ukupna suma verovatnoća bude jednaka nuli.

$$p(\mathbf{y}_k|\mathbf{y}_{1:k-1}) = \int p(\mathbf{y}_k|\mathbf{x}_k, \mathbf{y}_{1:k-1})p(\mathbf{x}_k|\mathbf{y}_{1:k-1})dx_k \quad (5.12)$$

Kompleksnost ovakvog izračunavanja *posteriorne* funkcije gustine raspodela verovatnoća, tokom vremena eksponenecijalno raste, zbog toga što se broj merenja uvećava tokom vremena [87].

Da bi se omogućilo izračunavanje, Bajesovi filteri prepostavljaju da je dinamički sistem opisan lancima Markova, što znači da trenutno stanje promenljive \mathbf{x}_k sadrži sve relevantne informacije.

Markovljeva pretpostavka implicira i da stanje u trenutku k zavisi samo od prethodnog stanja \mathbf{x}_{k-1} , $p(\mathbf{x}_k|\mathbf{x}_{k-1}, \mathbf{y}_{1:k-1}) = p(\mathbf{x}_k|\mathbf{x}_{k-1})$. Koristeći Markovljevu pretpostavku možemo efikasno sračunati *apriornu* funkciju gustine raspodele verovatnoće $p(\mathbf{x}_k|\mathbf{y}_{1:k-1})$ bez gubljenja infomacija.

$$p(\mathbf{x}_k|\mathbf{y}_{1:k-1}) = \int p(\mathbf{x}_k|\mathbf{x}_{k-1})p(\mathbf{x}_{k-1}|\mathbf{y}_{1:k-1}) dx_{k-1} \quad (5.13)$$

Ovde, $p(\mathbf{x}_k|\mathbf{x}_{k-1})$ opisuje dinamiku sistema, tj. kako se sistem menja u vremenu.

U trenutku k , merenje y_k postaje dostupno i koristimo ga kada korigujemo *a priori* funkciju gustine raspodele. Pozivajući se na Bajesovo pravilo, $p(x|y) = \frac{p(y|x)p(x)}{p(y)}$, izračunavanje *posteriorne* funkcije gustine raspodele može se uraditi na sledeći način.

$$p(\mathbf{x}_k|\mathbf{y}_{1:k}) = \frac{p(\mathbf{y}_k|\mathbf{x}_k)p(\mathbf{x}_k|\mathbf{y}_{1:k-1})}{p(\mathbf{y}_k|\mathbf{y}_{1:k-1})} \quad (5.14)$$

gde je normalizaciona konstanta definisana kao

$$p(\mathbf{y}_k|\mathbf{y}_{1:k-1}) = \int p(\mathbf{y}_k|\mathbf{x}_k)p(\mathbf{x}_k|\mathbf{y}_{1:k-1})d\mathbf{x}_k \quad (5.15)$$

zavisi od verovatnoće funkcije $p(\mathbf{y}_k|\mathbf{x}_k)$, definisane opservacionim modelom (5.9) i od poznatog šuma \mathbf{v}_k . U korekcionom koraku (5.14), merenje y_k se koristi da bi modifikovali *a priori* gustinu raspodele u cilju dobijanja tražene *posteriorne* funkcije gustine raspodele verovatnoće trenutnog stanja.

Jednačine (5.13) i (5.14) definišu rekurzivni Bajesov filter sa inicijalnim uslovom koji se definiše na osnovu specifične *a priori* funkcije gustine raspodele verovatnoća $p(\mathbf{y}_0) = p(\mathbf{x}_0)$, gde je \mathbf{y}_0 prazan skup.

5.1.1. Kalmanov filter

Kalmanov filter je linearni, stohastički, povratni, težinski, Bajesov estimator najmanjih kvadrata zasnovan na modelu [90]. On procenjuje stanje sistema, ili jednog njegovog dela, na osnovu znanja, odnosno na osnovu merenja ulaznih i izlaznih promenljivih sistema.

Ukoliko je $p(\mathbf{x}_0)$, $p(\mathbf{x}_k|\mathbf{x}_{k-1})$ i $p(\mathbf{y}_k|\mathbf{x}_k)$ Gausova funkcija raspodele, tada je i posteriorna gustina raspodele takođe Gausova, tj. važe sledeće pretpostavke [91]:

- 1) \mathbf{w}_k i \mathbf{v}_k se mogu dobiti na osnovu poznatih parametara Gausove raspodele.
- 2) $f_k(\mathbf{x}_{k-1}, \mathbf{w}_k)$ je poznata i linearna.
- 3) $h_k(\mathbf{x}_k, \mathbf{v}_k)$ je poznata i linearna.

Dakle, jednačina stanja i opservacije postaju linearane i Gausovske i mogu se napisati kao

$$\mathbf{x}_k = \mathbf{F}_k \mathbf{x}_{k-1} + \mathbf{w}_k \quad (5.16)$$

$$\mathbf{y}_k = \mathbf{H}_k \mathbf{x}_k + \mathbf{v}_k \quad (5.17)$$

gde je \mathbf{F}_k (po potrebi vremenski zavisna) prelazna matrica stanja koja određuje vezu između trenutnih i prethodnih stanja. \mathbf{B}_k je matrica upravljanja. Procesni šum \mathbf{w}_k je po pretpostavci beli šum, sa normalnom verovatnoćom raspodele:

$$p(\mathbf{w}_k) \sim N(0, \mathbf{Q}_k). \quad (5.18)$$

Njegova kovarijaciona matrica u trenutku k može se označiti sa \mathbf{Q}_k

$$\mathbf{Q}_k = E[\mathbf{w}_k \mathbf{w}_k^T] \quad (5.19)$$

Drugim rečima pretpostavlja se da je w_k beli šum sa nultom srednjom vrednošću sa poznatom matricom kovarijance \mathbf{Q}_k .

Kako se najčešće u realnom slučaju stanja sistema \mathbf{x}_k ne mogu direktno meriti, umesto njih se sensorima registruju izlazne vrednosti \mathbf{y}_k , koje su date sumom linearnih kombinacija promenljivih stanja $\mathbf{H}_k \mathbf{x}_k$, i šuma merenja \mathbf{v}_k .

Za šum merenja \mathbf{v}_k takođe se pretpostavlja da je nekorelisan i sa nultom srednjom vrednošću, kao i šum procesa w_k , i sa normalnom verovatnoćom raspodele

$$p(\mathbf{v}_k) \sim N(0, \mathbf{R}_k). \quad (5.20)$$

a njegova kovarijaciona matrica označava se sa \mathbf{R}_k

$$\mathbf{R}_k = E[\mathbf{v}_k \mathbf{v}_k^T] \quad (5.21)$$

Na osnovu modela može se pretpostaviti kakva će biti promena vrednosti vektora stanja između trenutaka $k-1$ i k i ovu procenu nazivamo *predikcijom* ili *a priori* estimacijom stanja i označavamo je sa $\hat{\mathbf{x}}_k^-$. Oznaka " - " u superskriptu stanja xoznačava da se ova procena generiše pre nego što je uzeto aktuelno merenje \mathbf{y}_k . Sa $\hat{\mathbf{x}}_k$ označava se *posteriori* stanje ili *estimirano (korigovano)* stanje koje se dobija posle ažuriranja na osnovu aktuelnog merenja \mathbf{y}_k .

Sada se *a priorna* i *posteriorna* grešku estimacije mogu definisati kao:

$$\mathbf{e}_k^- = \mathbf{x}_k - \hat{\mathbf{x}}_k^- \quad (5.22)$$

$$\mathbf{e}_k = \mathbf{x}_k - \hat{\mathbf{x}}_k \quad (5.23)$$

a-priori kovarijansa grešaka estimacije definišemo kao

$$\mathbf{P}_k^- = \mathbf{E}[\mathbf{e}_k^- \mathbf{e}_k^{-T}] \quad (5.24)$$

posteriorna kovarijansa grešaka estimacije jednaka je

$$\mathbf{P}_k = \mathbf{E}[\mathbf{e}_k \mathbf{e}_k^T] \quad (5.25)$$

Prilikom izvođenja jednačina za Kalmanov filter, počinje se s ciljem pronalaženja jednačine koja izračunava *posteriorino* stanje $\hat{\mathbf{x}}_k$ kao linearnu kombinaciju prediktovanog ili *a priori* stanja $\hat{\mathbf{x}}_k^-$ i težinske razlike između vrednosti stvarnog merenja \mathbf{y}_k (merenje dobijeno sa senzora) i predviđenog merenja $\mathbf{H}_k \hat{\mathbf{x}}_k^-$ koje se računa na osnovu *a priori* stanja $\hat{\mathbf{x}}_k^-$ [93][94].

$$\hat{\mathbf{x}}_k = \hat{\mathbf{x}}_k^- + \mathbf{K}_k (\mathbf{y}_k - \mathbf{H}_k \hat{\mathbf{x}}_k^-) \quad (5.26)$$

Razlika $(\mathbf{y}_k - \mathbf{H}_k \hat{\mathbf{x}}_k^-)$ se naziva rezidual. Rezidual odražava nesklad između predviđene vrednosti merenja i stvarne vrednosti merenja.

Matrica \mathbf{K}_k predstavlja Kalmanovo pojačanje koje minimizuje *posteriornu* kovarijacionu matricu greške estimacije. *Posteriorna* kovarijaciona matrica greške estimacije može se izvesti i na sledeći način [91]:

$$\mathbf{P}_k = \mathbf{P}_k^- - \mathbf{K}_k \mathbf{H}_k \mathbf{P}_k^- \quad (5.27)$$

Kalmanovo pojačanje \mathbf{K}_k se definiše kao:

$$\mathbf{K}_k = \mathbf{P}_k^- \mathbf{H}_k^T (\mathbf{H}_k \mathbf{P}_k^- \mathbf{H}_k^T + \mathbf{R}_k)^{-1} \quad (5.28)$$

gde je \mathbf{P}_k^- *a priori* kovarijaciona matrica greške estimacije i izvodimo je na sledeći način [91]:

$$\mathbf{P}_k^- = \mathbf{F}_k \mathbf{P}_{k-1} \mathbf{F}_k^T + \mathbf{Q}_k \quad (5.29)$$

Najjednostavniji slučaj kada je signal opisan modelom prvog reda u kojem je $\mathbf{H}_k = 1$; za svako k , a varijanse šumova procesa i merenja su q i r , respektivno. U ovom slučaju, izraz za Kalmanovo pojačanje postaje:

$$K_k = \frac{P_k^-}{P_k^- + r} \quad (5.30)$$

Iz (5.29) se vidi da je P_k^- srazmerno kovarijansi šuma merenja q , pa je :

$$K_k \approx \frac{q}{q+r} = \frac{1}{1+\frac{r}{q}} \quad (5.31)$$

Na osnovu prethodne jednačine vidi se da sa porastom varijanse šuma procesa q Kalmanovo pojačanje raste, dok ukoliko varijansa merenja r raste, Kalmanovo pojačanje se smanjuje. Iz jednačine korekcije procene stanja (5.26) vidi se da vrednost Kalmanovog pojačanja određuje u kojoj meri se estimacija zasniva na predikciji \hat{x}_k^- , a u kojoj meri na merenju, odnosno rezidualu merenja $y_k - x_k$ [92].

U slučaju kada $q \rightarrow \infty$, odnosno kada je šum procesa toliko veliki da se na dostupni model u prostoru stanja uopšte ne može osloniti, estimaciju treba generisati samo na osnovu reziduala merenja, što upravo i sledi iz izraza za Kalmanovo pojačanje:

$$q \rightarrow \infty \Rightarrow K_k \rightarrow 1 \Rightarrow \hat{x}_k = \hat{x}_k^- + (y_k - \hat{x}_k^-) = y_k \quad (5.32)$$

S druge strane, ukoliko je šum procesa q relativno mali, odnosno ukoliko model u velikoj meri odgovara realnom procesu, ali je šum merenja veliki, estimacija se vrši pretežno na osnovu modela. Merenja će biti potisnuta, što je i logično s obzirom na veliki sadržaj šuma u njima. U graničnom slučaju je

$$r \rightarrow \infty \Rightarrow K_k \rightarrow 0 \Rightarrow \hat{x}_k = \hat{x}_k^- + 0(y_k - \hat{x}_k^-) = F_{k-1} \hat{x}_{(k-1)} \quad (5.33)$$

Iz svega navedenog zaključuje se da Kalmanovo pojačanje ima jasnu interpretaciju faktora koji određuje uticaj svakog od izvora informacija (modela, odnosno merenja) na estimaciju. Pri tome je njegova vrednost jasno i logično određena odnosima varijansi šumova procesa i merenja [92].

Iz jednačina (5.16) i (5.17) vidi se da se Kalmanov filter sastoji od dva modela: model procesa i model merenja. Kalmanov filter procenjuje odnosno estimira proces na način povratne sprege. Prvo, procenjuje stanje u sistemu u nekom vremenskom trenutku, a zatim dobija povratnu informaciju u vidu merenja koje je zašumljeno. Prema tome, jednačine koje opisuju Kalmanov filter mogu se podeliti u dve grupe, jednačine predikcije i jednačine korekcije:

Prilikom inicijalizacije Kalmanovog filtera, početno stanje procesa \mathbf{x}_0 nije poznato, ali se pretpostavlja da su njegovo očekivanje $\hat{\mathbf{x}}_0^+ = \mathbf{m}_0 = \mathbf{E}[\mathbf{x}_0]$ i kovarijaciona matrica $\mathbf{P}_0^+ = \mathbf{E}[(\mathbf{x}_0 - \mathbf{m}_0)(\mathbf{x}_0 - \mathbf{m}_0)^T]$ dati [92].

Kao je već rečeno u slučaju kada vizija ne šalje validne podatke potrebno je da filter vrši predikciju pozicije čoveka. Kod Kalmanovog filtera, u ovoj tezi, pre koraka korekcije, dodat je još jedan korak koji ispituje da li nam sistem za vizije šalje validne podatke. Na osnovu ovoga i prethodno izvedenih jednačina moguće je sumirati opis toka algoritma koji vrši Kalmanovu filtraciju:

1) Inicijalizacija

Indeks k ima vrednost 0, i usvajaju se početne vrednosti procene vektora stanja i kovarijacione matrice greške estimacije

$$\hat{\mathbf{x}}_0^+ = \mathbf{m}_0 = \mathbf{E}[\mathbf{x}_0] \quad (5.34)$$

$$\mathbf{P}_0^+ = \mathbf{E}[(\mathbf{x}_0 - \mathbf{m}_0)(\mathbf{x}_0 - \mathbf{m}_0)^T] \quad (5.35)$$

2) Predikcija

$$\hat{\mathbf{x}}_k^- = \mathbf{F}_k \hat{\mathbf{x}}_{k-1} \quad (5.36)$$

$$\mathbf{P}_k^- = \mathbf{F}_k \mathbf{P}_{k-1} \mathbf{F}_k^T + \mathbf{Q}_k \quad (5.37)$$

3) Ispitivanje

Ukoliko vizija ne daje podatke, inkrementiranje indeksa k vraćanje na korak 2

4) Korekcija

$$\mathbf{K}_k = \mathbf{P}_k^- \mathbf{H}_k^T (\mathbf{H}_k \mathbf{P}_k^- \mathbf{H}_k^T + \mathbf{R}_k)^{-1} \quad (5.38)$$

$$\hat{\mathbf{x}}_k = \hat{\mathbf{x}}_k^- + \mathbf{K}_k (\mathbf{y}_k - \mathbf{H}_k \hat{\mathbf{x}}_k^-) \quad (5.39)$$

$$\mathbf{P}_k = \mathbf{P}_k^- - \mathbf{K}_k \mathbf{H}_k \mathbf{P}_k^- \quad (5.40)$$

Inkrementiranje indeksa k i skok na korak 2.

5.1.2. Prošireni Kalmanov filter

U slučaju kada je proces koji je potrebno estimirati opisan nelinearnim jednačinama ili je pak veza između procesa i meranja nelinearna, koristi se prošireni Kalmanov filter (*eng.* Extended Kalman filter), koji linearizuje estimacije koristeći parcijalne izvode procesa i funkcije kojim su opisana meranja. Linearni modeli imaju mnogo prednosti, laki su za računanje i za razumevanje. Pored toga, teorija linearizacije je potpuno razvijena i linearne diferencijalne jednačine je lako izračunati. Na osnovu toga razvijene su mnoge tehnike linearizacije nelinearnih modela, koje procenjuju stanje nelinearnih dinamičkih sistema linearizujući njihove nelinearnosti. Prošireni Kalmanov filter koristi jednačine standardnog Kalmanovog filtera prvog reda aproksimacije nelinearnog modela i linearizuje model u okolini trenutne procene stanja sistema. Prošireni Kalmanov filter se ubraja u

jednu od najpopularnijih tehnika estimacije stanja nelinearnih sistema i predstavlja efikasan metod za generisanje približne maksimalne verovatne procene stanja diskretnih nelinearnih dinamičkih sistema.

Kao što je već ranije rečeno, nelinearni dinamički sistem može se definisati sledećim jednačinama:

$$\mathbf{x}_k = f_k(\mathbf{x}_{k-1}, \mathbf{w}_k) \quad (5.41)$$

dok se stohastički opservacioni (mereni) proces opisuje sledećom jednačinom:

$$\mathbf{y}_k = h_k(\mathbf{x}_k, \mathbf{v}_k) \quad (5.42)$$

Nelinearna funkcija f povezuje prethodno stanje \mathbf{x}_{k-1} sa trenutnim \mathbf{x}_k i šumom procesa \mathbf{w}_k . Dok nelinearna funkcija h povezuje stanje \mathbf{x}_k i merenje \mathbf{y}_k kao i šum merenja \mathbf{v}_k .

U praksi individualne vrednosti šuma \mathbf{w}_k i \mathbf{v}_k nisu poznate u svakom vremenskom koraku [93].

Prema tome aproksimirani vektor stanja i merenja bez vrednosti sumaje:

$$\tilde{\mathbf{x}}_k = f_k(\hat{\mathbf{x}}_{k-1}, 0) \quad (5.43)$$

$$\tilde{\mathbf{y}}_k = h_k(\tilde{\mathbf{x}}_k, 0) \quad (5.44)$$

$\tilde{\mathbf{x}}_k$ i $\tilde{\mathbf{y}}_k$ predstavljaju aproksimirane vrednosti vektora stanja i vektora merenja, slično kao *a priori* stanje estimacije $\hat{\mathbf{x}}_k^-$ i predviđenog merenja $\mathbf{H}_k \hat{\mathbf{x}}_k^-$ kod Kalmanovog filtera. $\hat{\mathbf{x}}_k$ je *a posteriori* estimacija stanja u trenutku k .

Da bi se estimirao nelinearni proces prvo je potrebno napisati jednačine u obliku koji linearizuje jednačine (5.43) i (5.44) [94].

$$\mathbf{x}_k = \tilde{\mathbf{x}}_k + \mathbf{F}_k(\mathbf{x}_{k-1} - \hat{\mathbf{x}}_{k-1}) + \mathbf{W}_k \mathbf{w}_k \quad (5.45)$$

$$\mathbf{y}_k \approx \tilde{\mathbf{y}}_k + \mathbf{H}_k(\mathbf{x}_k - \tilde{\mathbf{x}}_k) + \mathbf{V}_k \mathbf{v}_k \quad (5.46)$$

gde su \mathbf{x}_k i \mathbf{y}_k stvarne vrednosti vektora stanja i vektora merenja.

\mathbf{F}_k predstavlja Jakobianovu matricu sa parcijalnim izvodima sistemske funkcije f u odnosu na stanje \mathbf{x} .

$$\mathbf{F}_k = \left. \frac{\partial f}{\partial \mathbf{x}} \right|_{(\hat{\mathbf{x}}_{k-1}, 0)} \quad (5.47)$$

\mathbf{W}_k predstavlja Jakobianovu matricu sa parcijalnim izvodima sistemske funkcije f u odnosu na beli šum \mathbf{w} .

$$\mathbf{W}_k = \left. \frac{\partial f}{\partial \mathbf{w}} \right|_{(\hat{\mathbf{x}}_{k-1}, 0)} \quad (5.48)$$

\mathbf{H}_k predstavlja Jakobianovu matricu sa parcijalnim izvodima sistemske funkcije h u odnosu na stanje \mathbf{x} .

$$\mathbf{H}_k = \left. \frac{\partial h}{\partial \mathbf{x}} \right|_{(\tilde{\mathbf{x}}_k, 0)} \quad (5.49)$$

\mathbf{V}_k predstavlja Jakobianovu matricu sa parcijalnim izvodima sistemske funkcije h u odnosu na beli šum \mathbf{v} .

$$\mathbf{V}_k = \left. \frac{\partial h}{\partial \mathbf{v}} \right|_{(\tilde{\mathbf{x}}_k, 0)} \quad (5.50)$$

Subskritk u Jakobijanovim matricama označava da su matrice raziličite u svakom koraku k , tj. parcijalni izvodi se izračunavaju u svakom koraku k .

Sada se greška predikcije i rezidual merenja definišu kao:

$$\tilde{\mathbf{e}}_{x_k} = \mathbf{x}_k - \tilde{\mathbf{x}}_k \quad (5.51)$$

$$\tilde{\mathbf{e}}_{y_k} = \mathbf{y}_k - \tilde{\mathbf{y}}_k \quad (5.52)$$

U realnom slučaju vrednost \mathbf{x}_k nije poznata, ali imamo pristup merenju \mathbf{y}_k . Koristeći jednačine (5.51) i (5.52), jednačinaza grešku procesa je [94]:

$$\tilde{\mathbf{e}}_{x_k} = \mathbf{F}_k(\mathbf{x}_{k-1} - \hat{\mathbf{x}}_{k-1}) + \varepsilon_k \quad (5.53)$$

$$\tilde{\mathbf{e}}_{y_k} = \mathbf{H}_k \tilde{\mathbf{e}}_{x_k} - \mu_k \quad (5.54)$$

gde ε_k i μ_k predstavljaju nove nezavisne promenljive sa nultom srednjom vrednošću i kovarijacionim matricama $\mathbf{W}_k \mathbf{Q}_k \mathbf{W}_k^T$ i $\mathbf{V}_k \mathbf{R}_k \mathbf{V}_k^T$. \mathbf{Q}_k i \mathbf{R}_k koje se računaju na osnovu jednačina (5.17) i (5.21) respektivno.

Jednačine (5.53) i (5.54) su linearne i najsličnijesu jednačinama modela procesa i jednačini modela merenja kod Kalmanovog filtera. Na osnovu ovog koristise rezidual aktuelnog merenja $\tilde{\mathbf{e}}_{y_k}$ u jednačini (5.52) da bi se estimirala greška predikcije $\tilde{\mathbf{e}}_{x_k}$ datu data jednačinom (5.53).

Ovu estimaciju označena sa $\hat{\mathbf{e}}_k$ i koristi se zajedno sa jednačinom (5.51) da bi se dobila *posteriorna* estimaciju stanja za originalni nelinearni proces.

$$\hat{\mathbf{x}}_k = \tilde{\mathbf{x}}_k - \hat{\mathbf{e}}_k \quad (5.55)$$

Verovatnoća raspodele promenljivih $\tilde{\mathbf{e}}_{x_k}$, ε_k i μ_k računa se približno kao:

$$p(\tilde{\mathbf{e}}_{x_k}) \sim N(0, \mathbf{E}[\tilde{\mathbf{e}}_{x_k}, \tilde{\mathbf{e}}_{x_k}^T]). \quad (5.56)$$

$$p(\varepsilon_k) \sim N(0, \mathbf{W}_k \mathbf{Q}_k \mathbf{W}_k^T). \quad (5.57)$$

$$p(\mu_k) \sim N(0, \mathbf{V}_k \mathbf{R}_k \mathbf{V}_k^T). \quad (5.58)$$

Na osnovu jednačina (5.53), (5.44) i (5.57) dobija se da je aproksimirana vrednost (*a-priori*) kovarijanse grešaka estimacije za prošireni Kalmanov filter jednaka:

$$\tilde{\mathbf{P}}_k = \mathbf{F}_k \mathbf{P}_{k-1} \mathbf{F}_k^T + \mathbf{W}_k \mathbf{Q}_k \mathbf{W}_k^T \quad (5.59)$$

Na osnovu ovih približnih raspodela uz pretpostavku da je *a priori* vrednost $\hat{\mathbf{e}}_k^-$ jednaka nuli, Kalmanova jednačina koja se koristi za estimaciju $\hat{\mathbf{e}}_k$ postaje:

$$\hat{\mathbf{e}}_k = \mathbf{K}_k \tilde{\mathbf{e}}_{y_k} \quad (5.60)$$

Zamenom jednačine (5.60) u jednačinu (5.55) i korišćenjem jednačine (5.52) dobija se

$$\hat{\mathbf{x}}_k = \tilde{\mathbf{x}}_k + \mathbf{K}_k \tilde{\mathbf{e}}_{y_k} = \tilde{\mathbf{x}}_k + \mathbf{K}_k (\mathbf{y}_k - \tilde{\mathbf{y}}_k) \quad (5.61)$$

Jednačina (5.61) se koristi kao jednačina korekcije u proširenom Kalmanovom filteru, u kojoj se $\tilde{\mathbf{x}}_k$ i $\tilde{\mathbf{y}}_k$ dobijamo dobijaju na osnovu jednačine (5.43) i (5.44).

Kalmanovo pojačanje se računa na osnovu jednačine (5.28) sa odgovarajućom zamenom za kovarijacionu matricu greške merenja tako da se dobija sledeća jednačina:

$$\mathbf{K}_k = \tilde{\mathbf{P}}_k \mathbf{H}_k^T (\mathbf{H}_k \tilde{\mathbf{P}}_k \mathbf{H}_k^T + \mathbf{V}_k \mathbf{R}_k \mathbf{V}_k^T)^{-1} \quad (5.62)$$

Korigovana (*a-posteriori*) kovarijansa grešaka estimacije jednaka je:

$$\mathbf{P}_k = \tilde{\mathbf{P}}_k - \mathbf{K}_k \mathbf{H}_k \tilde{\mathbf{P}}_k \quad (5.63)$$

Sada se konačno izvedene jednačine mogu sumirati kroz opis toka algoritma proširenog Kalmanovog filter. Kao i kod linearnog Kalmanovog filtera imamo još jedan korak, gde se proverava da li vizija šalje podatke.

1) Inicijalizacija

Indeks k ima vrednost 0, i usvajaju se početne vrednosti procene vektora stanja i kovarijacione matrice greške estimacije. Inicijalno, pošto su jedine dostupne informacije - matematičko očekivanje \mathbf{m}_0 i kovarijansa početnog stanja \mathbf{P}_0 , inicijalna ocena stanja i kovarijansa greške imaju oblik:

$$\hat{\mathbf{x}} = \mathbf{m}_0 = \mathbf{E}[\mathbf{x}_0] \quad (5.64)$$

$$\mathbf{P}_0 = \mathbf{E}[(\mathbf{x}_0 - \mathbf{m}_0)(\mathbf{x}_0 - \mathbf{m}_0)^T] \quad (5.65)$$

2) Predikcija

$$\tilde{\mathbf{x}}_k = f_k(\hat{\mathbf{x}}_{k-1}, 0) \quad (5.66)$$

$$\tilde{\mathbf{P}}_k = \mathbf{F}_k \mathbf{P}_{k-1} \mathbf{F}_k^T + \mathbf{W}_k \mathbf{Q}_k \mathbf{W}_k^T \quad (5.67)$$

3) Ispitivanje

Ukoliko vizija ne daje podatke, inkrementiranje indeksa k vraćanje na korak 2

4) Korekcija

$$\mathbf{K}_k = \tilde{\mathbf{P}}_k \mathbf{H}_k^T (\mathbf{H}_k \tilde{\mathbf{P}}_k \mathbf{H}_k^T + \mathbf{V}_k \mathbf{R}_k \mathbf{V}_k^T)^{-1} \quad (5.68)$$

$$\hat{\mathbf{x}}_k = \tilde{\mathbf{x}}_k + \mathbf{K}_k (\mathbf{y}_k - \tilde{\mathbf{y}}_k) \quad (5.69)$$

$$\mathbf{P}_k = \tilde{\mathbf{P}}_k - \mathbf{K}_k \mathbf{H}_k \tilde{\mathbf{P}}_k \quad (5.70)$$

Inkrementiranje indeksa k i skok na korak 2.

5.1.3. Partikle filter

Budući da kod mnogih praktičnih problema upotreba linearnih modela ili pretpostavka Gausovog šuma nije opravdana, postoji potreba za rekurzivnim tehnikama estimacija stanja za nelinearne i ne-gausovske probleme [88].

Razvojem i poboljšanjem računara došlo je i do sve veće primene Monte Karlo tehnike u istraživanjima. Partikle filter (*eng.* Particle filter) je prvi put predstavljen 1993. godine, i spada u klasu algoritama na osnovu sekvencijalnih Monte Karlo simulacija kao tehnike za rešavanje Bajesovskih problema estimacije.

Sekvencijalno uzorkovanje po značajnosti (*eng.* sequential importance sampling – SIS) je Monte Karlo metoda, osnova koja se koristi za većinu Partikle filtera i naziva se još i sekvencijalni Monte Karlo (*eng.* Sequential Monte Carlo -SMC). Kod ovog Partikle filtera odabiranje čestica-partikala vrši se sekvencijalno na osnovu težina,[95][96].

Ovaj SMC pristup se takođe može naći u literaturi podrazličitim imenima [97]. Poznat je i kao “bootstrap” filter (*eng.* bootstrap filtering) [98], kondenzacioni algoritam (*eng.* condensation algorithm), [99], Partikle filter [100], interakcija aproksimiranih čestica (*eng.* interacting particle approximations)[101] i opstanak najsposobnije (*eng.* survival of the fittest) [102].

Može se reći da je sekvencijalni Monte Karlo metoda tehnika za implementaciju rekurzivnog Bajesovog filtera uz pomoć MC simulacija. Osnovna ideja je da setražena funkcija gustine verovatnoće predstavi skupom slučajnih uzoraka sa pridruženim težinama i da se izračunaju estimacije na osnovu ovih uzoraka i težina. Kako broj uzoraka postaje veći, Monte Karlo karakterizacija postaje ekvivalentna reprezentacija uobičajenoj funkcionalnoj deskripciji *posteriorene* funkcije gustine verovatnoće i SIS filterse približava Bajesovoj estimaciji.

Različite verzije Partikle filtera koje su predložene u literaturi mogu se smatrati specijalnim slučajevima osnovnog SIS algoritma.

Sekvencijalno uzorkovanje sa resamplovanjem (*eng.* sequential importance resampling – SIR), razvijeno je odvojeno od SIS algoritma od strane Gordon-a i drugih [98]. SIR sadrži korak ponovnog uzorkovanja ili resamplovanja (*eng.* resampling) u svakoj iteraciji sekvencijalnog uzorkovanja po značajnosti. Od tada je dokazano da resamplovanje ima značajnu praktičnu i teorijsku prednost [95].

Takođe, postoji i pomoćni Partikle filter (*eng.* auxiliary particle filter - APF) kod koga je osnovna ideja da se poveća uticaj čestica-partikala s velikom verovatnoćom, pomoću dodatnog indeksa za svaku česticu, tako da se može pratiti poreklo čestica, dok se verovatnoća procenjuje na sledećem vremenskom koraku, pod pretpostavkom da estimacija nije potrebna do sledećeg merenja [103].

Pored ovih se pominju još i marginalizovani Partikle filter (*eng.* marginalized particle filter - MPF), mada se ideja marginalizacije ponekad naziva i Rao-Blackwell-izacija (*eng.* Rao-Blackwellized particle filter - RBPF), [104], [105].

Treba napomenuti još da se svi ovi specijalni slučajevi mogu izvesti iz SIS algoritma odgovarajućim izborom gustine uzorkovanja čestica na osnovu težina i/ili modifikacije koraka u kom vršimo resamplovanje (*eng.* resampling).

5.1.3.1. Monte Karlo

Monte Carlo metoda se primenjuje u raznim simulacijama koje koriste slučajne brojeve, a takve simulacije su označene kao Monte Carlo simulacije. Najčešće se ovaj termin koristi samo za statičke tipove simulacija kod kojih se u rešavanju problema koristi stvaranje uzoraka iz raspodela slučajnih promenljivih. Pri tome, problemi mogu biti bilo determinističkog, bilo stohastičkog karaktera.

Kod Bajesovih filtera, glavni problem može se često redukovati kao izračunavanje matematičkog očekivanja *posteriorne* funkcije gustine raspodele [106]:

$$E[g(\mathbf{x}_k)|\mathbf{y}_{1:k}] = \int g(\mathbf{x}_k)p(\mathbf{x}_k|\mathbf{y}_{1:k}) d\mathbf{x}_k \quad (5.71)$$

gde je $g : \mathbb{R}^n \rightarrow \mathbb{R}^m$ proizvoljna funkcija, a $p(\mathbf{x}_k|\mathbf{y}_{1:k})$ *posteriorna* funkcija gustine verovatnoće stanja \mathbf{x}_k za data merenja y_1, \dots, y_k .

Sada je problem sveden na problem izračunavanja integrala, koji može biti rešen primenom Monte Karlo metode.

Kod Monte Karlo aproksimacije, *posteriornu* funkciju gustinu raspodele verovatnoće $p(\mathbf{x}_k|\mathbf{y}_{1:k})$ aproksimiramo skupom od nasumičnih N čestica-partikla

$$\mathbf{x}_k^{(i)} \sim p(\mathbf{x}_k|\mathbf{y}_{1:k}), \quad i = 1, \dots, N. \quad (5.72)$$

i estimiramo očekivanje kao

$$E[g(\mathbf{x}_k)|\mathbf{y}_{1:k}] \approx \frac{1}{N} \sum_{i=1}^N g(\mathbf{x}_k^{(i)}) \quad (5.73)$$

Na ovaj način, Monte Karlo metoda aproksimira traženu raspodelu (*eng.* target distribution) skupom od N uzoraka koji su distribuirani prema traženoj gustini raspodele [106].

Konvergencija Monte Karlo aproksimacije je garantovana centralnom graničnom teoremom, pri čemu srednje kvadratna greška aproksimacije iznosi $O(N^{-1/2})$, bez obzira na dimenziju promeljive x_k .

5.1.3.2 . Uzorkovanje po značajnosti

Često u realnom slučaju nije moguće dobiti uzorke (*eng. samples*) direktno iz gustine raspodele verovatnoće $p(\mathbf{x}_k|\mathbf{y}_{1:k})$ zbog toga što ona može biti nestandardna, multivarijaciona ili multimodalna [107]. Da bi prevazišli ovaj problem, koristimo princip *uzrokovanja po značajnosti*. Kod uzrokovanje po značajnosti, koristimo približnu gustinu raspodele verovatnoće koju nazivamo značajna raspodela (*eng. importance distribution*) ili predložena raspodela, $\pi(\mathbf{x}_k|\mathbf{y}_{1:k})$, iz koje je moguće lako izvući uzorke, pri čemu je $p(\mathbf{x}_k|\mathbf{y}_{1:k})$ proporcionalna $\pi(\mathbf{x}_k|\mathbf{y}_{1:k})$ za svako \mathbf{x}_k .

Budući da je $p(\mathbf{x}_k|\mathbf{y}_{1:k})$ normalizovana funkcija gustine raspodele, sledi da $\pi(\mathbf{x}_k|\mathbf{y}_{1:k})$ mora biti skalirana nenormalizovana verzija raspodele $p(\mathbf{x}_k|\mathbf{y}_{1:k})$.

Prema tome faktor skaliranja ili težina jednak je:

$$\xi(\mathbf{x}_k) = \frac{p(\mathbf{x}_k|\mathbf{y}_{1:k})}{\pi(\mathbf{x}_k|\mathbf{y}_{1:k})} \quad (5.74)$$

Uzrokovanje po značajnosti je zasnovano na dekompoziciji matematičkog očekivanja posterirone funkcije gustine raspodele $p(\mathbf{x}_k|\mathbf{y}_{1:k})$, primenom formule (5.71) gde dobijamo da je [106] :

$$E[g(\mathbf{x}_k)|\mathbf{y}_{1:k}] = \int g(\mathbf{x}_k)p(\mathbf{x}_k|\mathbf{y}_{1:k}) d\mathbf{x}_k = \int g(\mathbf{x}_k) \frac{p(\mathbf{x}_k|\mathbf{y}_{1:k})}{\pi(\mathbf{x}_k|\mathbf{y}_{1:k})} \pi(\mathbf{x}_k|\mathbf{y}_{1:k}) d\mathbf{x}_k \quad (5.75)$$

gde je $\pi(\mathbf{x}_k|\mathbf{y}_{1:k}) > 0$ kada je $p(\mathbf{x}_k|\mathbf{y}_{1:k}) > 0$.

Estimirano očekivanje dobijamo primenom Monte Karlo metode tako što predloženu funkciju gustine raspodele verovatnoće $\pi(\mathbf{x}_k|\mathbf{y}_{1:k})$ aproksimiramo skupom od nasumičnih N čestica-partikla:

$$\mathbf{x}_k^{(i)} \sim \pi(\mathbf{x}_k|\mathbf{y}_{1:k}), \quad i = 1, \dots, N. \quad (5.76)$$

Tada estimirano očekivanje iznosi:

$$E[g(\mathbf{x}_k)|\mathbf{y}_{1:k}] \approx \frac{1}{N} \sum_{i=1}^N \frac{p(\mathbf{x}_k^{(i)}|\mathbf{y}_{1:k})}{\pi(\mathbf{x}_k^{(i)}|\mathbf{y}_{1:k})} g(\mathbf{x}_k^{(i)}) = \frac{1}{N} \sum_{i=1}^N \xi(\mathbf{x}_k^{(i)}) g(\mathbf{x}_k^{(i)}) \quad (5.77)$$

gde su težine definisane kao:

$$\xi(\mathbf{x}_k)^{(i)} = \frac{1}{N} \frac{p(\mathbf{x}_k^{(i)} | \mathbf{y}_{1:k})}{\pi(\mathbf{x}_k^{(i)} | \mathbf{y}_{1:k})} \quad (5.78)$$

Kako uzorkujemo predloženu raspodelu, koja predstavlja aproksimaciju ciljane raspodele, raspodela uzoraka nije egzaktna, pa je stoga neophodno da je korigujemo pridružujući težine svakom uzorku [106].

Mana direktnog uzrokovanja po značajnosti je u tome što moramo da izračunamo gustinu raspodele $p(\mathbf{x}_k^{(i)} | \mathbf{y}_{1:k})$ da bi mogli da je koristimo. Pozivajući se na Bajesovo pravilo, izračunavamo *posteriornu* funkciju gustine raspodele na sledeći način:

$$p(\mathbf{x}_k^{(i)} | \mathbf{y}_{1:k}) = \frac{p(\mathbf{y}_{1:k} | \mathbf{x}_k^{(i)}) p(\mathbf{x}_k^{(i)})}{p(\mathbf{y}_{1:k})} \quad (5.79)$$

gde je $p(\mathbf{x}_k^{(i)})$ a *priorna* raspodela definisana dinamičkim modelom,

$p(\mathbf{y}_{1:k} | \mathbf{x}_k^{(i)})$ model verovatnoće za merenja, a

$p(\mathbf{y}_{1:k})$ normalizaciona konstanta

$$p(\mathbf{y}_{1:k}) = \int p(\mathbf{y}_{1:k} | \mathbf{x}_k) p(\mathbf{x}_k) d\mathbf{x}_k \quad (5.80)$$

Funkcije gustine $p(\mathbf{y}_{1:k} | \mathbf{x}_k^{(i)})$ i $p(\mathbf{x}_k^{(i)})$ je lako izračunati, ali je često veoma teško izračunati integral-normalizacionu konstantu u prethodnom izrazu.

Da bi se prevazisao ovaj problem može se ponovo primeniti dekompozicija matematičkog očekivanja i Bajesovo pravilo u kome će se uzorkovati i normalizaciona konstanta [106].

$$\begin{aligned}
E[g(\mathbf{x}_k)|\mathbf{y}_{1:k}] &= \int g(\mathbf{x}_k)p(\mathbf{x}_k|\mathbf{y}_{1:k}) d\mathbf{x}_k = \frac{\int g(\mathbf{x}_k)p(\mathbf{y}_{k:1}|\mathbf{x}_k)p(\mathbf{x}_k)d\mathbf{x}_k}{\int p(\mathbf{y}_{1:k}|\mathbf{x}_k)p(\mathbf{x}_k)d\mathbf{x}_k} \\
&= \frac{\int \left[g(\mathbf{x}_k) \frac{p(\mathbf{y}_{k:1}|\mathbf{x}_k)p(\mathbf{x}_k)}{\pi(\mathbf{x}_k|\mathbf{y}_{1:k})} \right] \pi(\mathbf{x}_k|\mathbf{y}_{1:k})d\mathbf{x}_k}{\int \left[\frac{p(\mathbf{y}_{k:1}|\mathbf{x}_k)p(\mathbf{x}_k)}{\pi(\mathbf{x}_k|\mathbf{y}_{1:k})} \right] \pi(\mathbf{x}_k|\mathbf{y}_{1:k})d\mathbf{x}_k} \\
&\approx \frac{\frac{1}{N} \sum_{i=1}^N \frac{p(\mathbf{y}_{1:k}|\mathbf{x}_k^{(i)})p(\mathbf{x}_k^{(i)})}{\pi(\mathbf{x}_k^{(i)}|\mathbf{y}_{1:k})} g(\mathbf{x}_k^{(i)})}{\frac{1}{N} \sum_{j=1}^N \frac{p(\mathbf{y}_{1:k}|\mathbf{x}_k^{(j)})p(\mathbf{x}_k^{(j)})}{\pi(\mathbf{x}_k^{(j)}|\mathbf{y}_{1:k})}} \\
&= \sum_{i=1}^N \left[\frac{\frac{p(\mathbf{y}_{1:k}|\mathbf{x}_k^{(i)})p(\mathbf{x}_k^{(i)})}{\pi(\mathbf{x}_k^{(i)}|\mathbf{y}_{1:k})}}{\frac{1}{N} \sum_{j=1}^N \frac{p(\mathbf{y}_{1:k}|\mathbf{x}_k^{(j)})p(\mathbf{x}_k^{(j)})}{\pi(\mathbf{x}_k^{(j)}|\mathbf{y}_{1:k})}} \right] g(\mathbf{x}_k^{(i)}) \\
&\quad \xi^{(i)}
\end{aligned} \tag{5.81}$$

Prema tome, za dati opservacioni model $p(\mathbf{y}_{k:1}|\mathbf{x}_k)$ i *a priori* funkciju gustine raspodele $p(\mathbf{x}_k)$ može se formirati aproksimaciju *posteriorne* funkcije $p(\mathbf{x}_k|\mathbf{y}_{1:k})$, primenom uzorkovanja po značajnosti na sledeći način [106]:

1. *Gustinu raspodele verovatnoće* $\pi(\mathbf{x}_k|\mathbf{y}_{1:k})$ *aproksimiramo skupom od nasumičnih* N *čestica-partikla*

$$\mathbf{x}_k^{(i)} \sim \pi(\mathbf{x}_k|\mathbf{y}_{1:k}), \quad i = 1, \dots, N. \tag{5.82}$$

2. *Izračunavanje normalizovane težine*

$$\xi^{(i)} = \frac{p(\mathbf{y}_{1:k}|\mathbf{x}_k^{(i)})p(\mathbf{x}_k^{(i)})}{\pi(\mathbf{x}_k^{(i)}|\mathbf{y}_{1:k})} \tag{5.83}$$

3. *Izračunavanje normalizovane težine*

$$\tilde{\xi}^{(i)} \approx \frac{\xi^{(i)}}{\sum_{i=1}^N \xi^{(i)}} \tag{5.84}$$

4. *Aproksimativna vrednost posteriorne funkcije gustine verovatnoće može se formalno napisati kao:*

$$p(\mathbf{x}_k|\mathbf{y}_{1:k}) \approx \sum_{i=1}^N \tilde{\xi}^{(i)} \delta(\mathbf{x}_k - \mathbf{x}_k^{(i)}) \tag{5.85}$$

5.1.3.3 . Sekvencijalno uzorkovanje po značajnosti –SIS metod

Sekvencijano uzorkovanje po značajnosti predstavlja sekvencijalnu verziju uzorkovanja po značajnosti [106]. Da bi se razvio algoritam, pretpostavka je da postoji skup od N težinskih čestica-uzoraka $\{x_k^i, \xi_k^i\}_{i=1}^N$, dobijenih iz predložene raspodele $\pi(\mathbf{x}_k | \mathbf{y}_{1:k})$ i koji predstavljaju funkciju gustine raspodele $p(\mathbf{x}_k | \mathbf{y}_{1:k-1})$ i za koje važe jednačine (5.77) i (5.85).

Takođe, razmatra se cela *posteriorna* funkcija gustine raspodele verovatnoće stanja $\mathbf{x}_{0:k}$ za data merenja $\mathbf{y}_{1:k}$. Na osnovu Markovljevihih pretpostavki, dobija se sledeća rekurzija za *posteriornu* raspedelu:

$$\begin{aligned} p(\mathbf{x}_{0:k} | \mathbf{y}_{1:k}) &\propto p(\mathbf{y}_k | \mathbf{x}_{0:k}, \mathbf{y}_{1:k-1}) p(\mathbf{x}_{0:k} | \mathbf{y}_{1:k-1}) \\ &= p(\mathbf{y}_k | \mathbf{x}_k) p(\mathbf{x}_k | \mathbf{x}_{0:k-1}, \mathbf{y}_{1:k-1}) p(\mathbf{x}_{0:k-1} | \mathbf{y}_{1:k-1}) \\ &= p(\mathbf{y}_k | \mathbf{x}_k) p(\mathbf{x}_k | \mathbf{x}_{0:k-1}) p(\mathbf{x}_{0:k-1} | \mathbf{y}_{1:k-1}) \end{aligned} \quad (5.86)$$

Koristeći sličnu analogiju, bira se predložena gustina raspodele tako da se može razložiti na sledeći način:

$$\pi(\mathbf{x}_{0:k} | \mathbf{y}_{1:k}) = \pi(\mathbf{x}_k | \mathbf{x}_{0:k-1}, \mathbf{y}_{1:k}) \pi(\mathbf{x}_{0:k-1} | \mathbf{y}_{1:k-1}) \quad (5.87)$$

I tada izraz za težinske koeficijente postaje:

$$\begin{aligned} \xi_k^{(i)} &\propto \frac{p(\mathbf{y}_k | \mathbf{x}_k^{(i)}) p(\mathbf{x}_k^{(i)} | \mathbf{x}_{k-1}^{(i)}) p(\mathbf{x}_{0:k-1}^{(i)} | \mathbf{y}_{1:k-1})}{\pi(\mathbf{x}_{0:k}^{(i)} | \mathbf{y}_{1:k})} \\ &= \frac{p(\mathbf{y}_k | \mathbf{x}_k^{(i)}) p(\mathbf{x}_k^{(i)} | \mathbf{x}_{k-1}^{(i)}) p(\mathbf{x}_{0:k-1}^{(i)} | \mathbf{y}_{1:k-1})}{\pi(\mathbf{x}_k^{(i)} | \mathbf{x}_{0:k-1}^{(i)}, \mathbf{y}_{1:k}) \pi(\mathbf{x}_{0:k-1}^{(i)} | \mathbf{y}_{1:k-1})} \end{aligned} \quad (5.88)$$

Pretpostavka je da već postoje izvučeni uzorci $\mathbf{x}_{0:k-1}^{(i)}$ iz predložene raspodele $\pi(\mathbf{x}_{0:k-1}^{(i)} | \mathbf{y}_{1:k-1})$ i izračunati odgovarajući težinski koeficijenti $\xi_{k-1}^{(i)}$.

$$\xi_{k-1}^{(i)} \propto \frac{p(\mathbf{x}_{0:k-1}^{(i)} | \mathbf{y}_{1:k-1})}{\pi(\mathbf{x}_{0:k-1}^{(i)} | \mathbf{y}_{1:k-1})} \quad (5.89)$$

Sada se mogu izvući čestice $\mathbf{x}_{0:k}^{(i)}$ iz predložene raspodele $(\mathbf{x}_{0:k} | \mathbf{y}_{1:k})$, uzorkujući samo nove čestice za korak k kao $\mathbf{x}_k^{(i)} \sim \pi(\mathbf{x}_k | \mathbf{x}_{0:k-1}^{(i)}, \mathbf{y}_{1:k})$.

$$\xi_k^{(i)} \propto \xi_{k-1}^{(i)} \frac{p(\mathbf{y}_k | \mathbf{x}_k^{(i)}) p(\mathbf{x}_k^{(i)} | \mathbf{x}_{k-1}^{(i)})}{\pi(\mathbf{x}_k^{(i)} | \mathbf{x}_{0:k-1}^{(i)}, \mathbf{y}_{1:k})} \quad (5.90)$$

Ukoliko se uzme da je predložena raspodela Markovljeva, što se i najčešće radi kod praktične primene, tada predložena raspodela zavisi samo od prethodnog stanja \mathbf{x}_{k-1} i prethodnog merenja \mathbf{y}_{k-1} :

$$\pi(\mathbf{x}_k | \mathbf{x}_{0:k-1}, \mathbf{y}_{1:k}) = \pi(\mathbf{x}_k | \mathbf{x}_{k-1}, \mathbf{y}_k) \quad (5.91)$$

Ovo je naročito korisno u opštem slučaju kada je potrebna samo estimirana vrednost $p(\mathbf{x}_k | \mathbf{y}_{1:k})$ u svakom koraku. Prema tome, računanje težinskih koeficijenata se modifikuje na sledeći način:

$$\xi_k^{(i)} \propto \xi_{k-1}^{(i)} \frac{p(\mathbf{y}_k | \mathbf{x}_k^{(i)}) p(\mathbf{x}_k^{(i)} | \mathbf{x}_{k-1}^{(i)})}{\pi(\mathbf{x}_k^{(i)} | \mathbf{x}_{k-1}^{(i)}, \mathbf{y}_k)} \quad (5.92)$$

i oni se normalizuju tako da suma svih težina bude jednaka jedinici:

$$\tilde{\xi}^{(i)} \approx \frac{\xi^{(i)}}{\sum_{j=1}^N \xi^{(j)}} \quad (5.93)$$

A *posteriona* gustina $p(\mathbf{x}_k | \mathbf{y}_{1:k})$ se može aproksimirati kao:

$$p(\mathbf{x}_k | \mathbf{y}_{1:k}) \approx \sum_{i=1}^N \tilde{\xi}^{(i)} \delta(\mathbf{x}_k - \mathbf{x}_k^{(i)}) \quad (5.94)$$

Gde se težinski koeficijenti računaju prema jednačini (5.92) i (5.93).

Sada se izvedene jednačine mogu sumirati kroz opis toka SIS algoritma [106]:

1. Inicijalizacija

a) Uzorkovanje čestice $\mathbf{x}_0^{(i)}$ iz *a priori* funkcije gustine raspodele

$$\mathbf{x}_0^{(i)} \sim p(\mathbf{x}_0), \quad i = 1, \dots, N. \quad (5.95)$$

b) izračunavanje težinskih koeficijenata

$$\xi_0^{(i)} = p(\mathbf{y}_0 | \mathbf{x}_0), \quad i = 1, \dots, N. \quad (5.96)$$

c) normalizacija težinskih koeficijenata.

$$\tilde{\xi}_0^{(i)} = \frac{\xi_0^{(i)}}{\sum_{j=1}^N \xi_0^{(j)}} \quad (5.97)$$

2. Predikcija

Uzorkovanje čestice $\mathbf{x}_k^{(i)}$ iz predložene funkcije gustine raspodele $\pi(\mathbf{x}_k | \mathbf{x}_{k-1}, \mathbf{y}_k)$.

$$\mathbf{x}_k^{(i)} \sim \pi(\mathbf{x}_k | \mathbf{x}_{k-1}, \mathbf{y}_k) \quad i = 1, \dots, N. \quad (5.98)$$

3. Korekcija

a) Izračunati nove težinske koeficijente prema jednačini (5.92)

b) Normalizovati težinske koeficijente prema jednačini (5.93).

Inkrementiranje indeksa k i skok na korak 2.

Čest problem sa SIS filterom je fenomen degeneracije, gde nakon nekoliko iteracija, težine svih čestica osim jedne imaju vrednost nula ili približno nuli [107]. Problem degeneracije se ogleda u velikom izračunavanju koje je posvećeno ažuriranju čestica čiji je doprinos u određivanju približne vrednosti, približno jednak nuli.

Ovaj problem može biti uspešno rešen korišćenjem metode resamplovanja koja će biti detaljnije objašnjena u narednom poglavlju.

5.1.3.4. Sekvencijalno uzorkovanje sa resamplovanjem-SIR metod

SIR metod je razvijen nezavisno od SIS algoritma od strane Gordona i drugih [98] i za razliku od SIS algoritma sadrži i još jedan korak, resamplovanje. Ideja resamplovanja je da ukloni čestice sa veoma malim težinskim koeficijentima i da multiplicira čestice sa većim težinskim koeficijentima. Često se u literaturi upravo SIR metod naziva Partikle filterom. Korak resamplovanja uključuje uzorkovanje novog seta čestica $\{\tilde{\mathbf{x}}_k^i\}_{i=1}^N$, koje aproksimiraju diskretnu gustinu raspodele $p(\mathbf{x}_k | \mathbf{y}_{1:k})$ definisanu težinskim koeficijentima (jednačina 5.94) i vrše zamenu starog skupa čestica novim.

Procedura resamplovanja se izvodi na sledeći način. Prvo, svaki težinski koeficijent se definiše kao verovatnoća dobijanja uzorka indeksa i u setu $\{\mathbf{x}_k^i\}_{i=1}^N$, zatim se uzorkuje novi skup čestica iz diskretne raspodele date jednačinom (5.94) i zameni stari skup čestica novim. Na kraju se podesi da sve težine imaju konstantnu vrednost $\xi^{(i)} = 1/N$.

Takođe, kod SIR algoritma se veoma često korak resamplovanja ne izvodi u svakom koraku, već samo kada je to potrebno [106]. Razlog za primenu ovog metoda leži u činjenici da kod resamplovanja u svakoj iteraciji, kada imamo veliki broj iteraciji, simulirane čestice su u interakciji i prestaju biti statistički nezavisne. Takođe, dolazi i do gubitka raznolikosti čestica koje prolaze kroz resamplovanje.

Kod adaptivnog resamplovanja, efektivan broj čestica, koji se procenjuje na osnovu varijanse težinskih koeficijenata, se koristi za praćenje potrebe za resamplovanjem. Procenjena vrednost efektivnog broja čestica se može izračunati kao:

$$n_{eff} \approx \frac{1}{\sum_{i=1}^N (\xi_k^{(i)})^2} \quad (5.99)$$

Resamplovanje se izvodi u slučaju kada je efektivni broj čestica značajno manji od ukupnog broja [106].

Za estimiranje pozicije čoveka na osnovu informacija dobijih sensorom robotske vizije, korišćen je SIR algoritam sa korakom ispitivanja, gde se proverava da li sensor robotske vizije šalje validne podatke.

Opis toka algoritma:

1. Inicijalizacija

- a) Uzorkovanje čestice $\mathbf{x}_0^{(i)}$ iz *a priori* funkcije gustine raspodele,

$$\mathbf{x}_0^{(i)} \sim p(\mathbf{x}_0), \quad i = 1, \dots, N. \quad (5.100)$$

- b) izračunavanje težinskih koeficijenata,

$$\xi_0^{(i)} = p(\mathbf{y}_0 | \mathbf{x}_0), \quad i = 1, \dots, N. \quad (5.101)$$

- c) normalizacija težinskih koeficijenata.

$$\tilde{\xi}_0^{(i)} = \frac{\xi_0^{(i)}}{\sum_{j=1}^N \xi_0^{(j)}} \quad (5.102)$$

2. Predikcija

Uzorkovanje čestice $\mathbf{x}_k^{(i)}$ iz predložene funkcije gustine raspodele $\pi(\mathbf{x}_k | \mathbf{x}_{k-1}, \mathbf{y}_k)$.

$$\mathbf{x}_k^{(i)} \sim \pi(\mathbf{x}_k | \mathbf{x}_{k-1}, \mathbf{y}_k) \quad i = 1, \dots, N. \quad (5.103)$$

3. Ispitivanje

Ukoliko vizija ne daje validne podatke sledi inkrementiranje indeksa k i skok na korak 2.

4. Korekcija

a) Izračunavanje nove težinske koeficijente prema jednačini

$$\xi_k^{(i)} \propto \xi_{k-1}^{(i)} \frac{p(\mathbf{y}_k | \mathbf{x}_k^{(i)}) p(\mathbf{x}_k^{(i)} | \mathbf{x}_{k-1}^{(i)})}{\pi(\mathbf{x}_k^{(i)} | \mathbf{x}_{k-1}^{(i)}, \mathbf{y}_k)} \quad (5.104)$$

b) Normalizovanje težinskih koeficijenata prema jednačini

$$\tilde{\xi}^{(i)} \approx \frac{\xi^{(i)}}{\sum_{j=1}^N \xi^{(j)}} \quad (5.105)$$

c) Ako je n_{eff} mnogo manje od N sledi korak resamplovanja

5. Resamplovanje

- Definisati $\tilde{\xi}^{(i)}$ kao verovatnoću dobijanja uzoraka indeksa i u setu $\{\mathbf{x}_k^i\}_{i=1}^N$.
- Uzorkovati uzorke, $\{\tilde{\mathbf{x}}_k^i\}_{i=1}^N$, iz diskretne raspodele i zameniti stari set uzoraka novim.
- Podesiti da sve težine imaju konstantnu vrednost $\tilde{\xi}^{(i)} = 1/N$.

Inkrementiranje indeksa k i skok na korak 2.

6. EKSPERIMENTALNI REZULTATI PRAĆENJA ČOVEKA NA OSNOVU PODATAKA DOBIJENIM SENZOROM ROBOTSKE VIZIJE

Razvijeni algoritmi za estimaciju položaja čoveka kojeg treba pratiti mobilnim robotom testirani su na Institutu za Automatiku (IAT) Univerziteta u Bremenu primenom sistema stereo vizije za detekciju čoveka koji koristi Point Grey Bumblebee XB3 stereo kameru [110] i na Mašinskom fakultetu Univerziteta u Nišu primenom sistema vizije koji koristi OpenNI biblioteku [12] i ASUS Xtion 3D senzor [11].

6.1. MOBILNA ROBOTSKA PLATFORMA NEOBOTIX MP-M470 SA INTEGRISANIM SISTEMOM STEREO VIZIJE

Prva korišćena mobilna robotska platforma ima integrisan sistem stereo vizije za praćenje čoveka koji je testiran u okviru radnog scenarija mobilnog robota sa namerom da prati ljudskog saradnika u zatvorenom prostoru (*eng. indoor*), kao i na otvorenom prostoru (*eng. outdoor*). Robotska platforma, razvijena na IAT-u [111], jeste mobilni robot MP-M470 (Slika 6.1), proizvedena od strane Neobotix GmbH-a, koja je opremljena višenamenskim 2D bezbednosnim laserom S3000 proizvođača SICK i Point Grey Bumblebee XB3 stereo kamerom. U tabeli 6.1 data je specifikacija Point Grey Bumblebee XB3 stereo kamere.



Slika 6.1. Mobilna robotska platforma Neobotix MP-M470 sa integrisanim sistemom stereo vizije

Tabela 6.1: Specifikacije Point Grey Bumblebee XB3 stereo kamere [110]

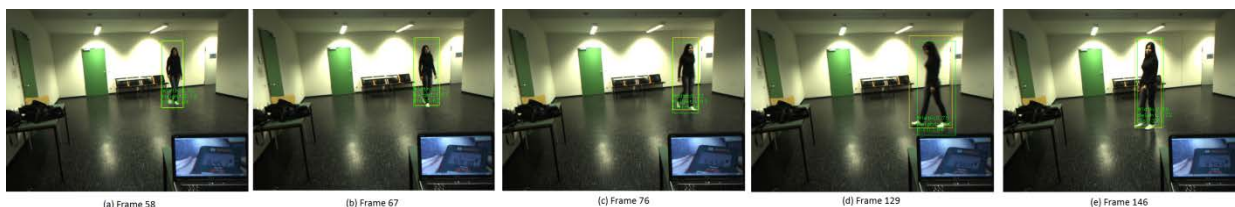
| | | | |
|-------------------------|-------------------------|---------------|---|
| Broj frejmova u sekundi | 16 fps | Dimenzije | 277 x 37 x 41.8mm |
| Senzor | Sony ICX445 | Fleš memorija | 512 KB |
| Rezolucija | 1280x960 , 1.3 MP | Masa | 505 g |
| Veličina piksela | 3.75 μm | Softver | Bumblebee XB3 Development Accessory Kit |
| Žižna daljina | 3.8 mm, 66-deg HFOV | Interfejs | FireWire 1394b |
| Operativni sistem | Windows ili Linux 32/64 | Platforma | Intel X86 & AMD |

Ekperimenti su izvođeni tako da čovek hoda ispred mobilnog robota na kome je postavljena Point Grey Bumblebee XB3 stereo kamera [110]. U eksperimentima, robot je bio statičan tako da je samo posmatrao osobu bez da je prati. Parovi slika dobijeni ovom kamerom su u punoj rezoluciji (1280 x 960) sa 12 frejmova u sekundi. Svaki par stereo slika je obrađen kako bi se dobile informacije za rekonstrukciju položaja čoveka baziranoj na stereo viziji u odnosu na koordinatni sistem kamere. Proces akvizicije, segmentacije i klasifikacije je detaljno objašnjen u poglavlju 2 ove teze.

Neki od primera obrađenih slika sa detektovanim i uokvirenim čovekom su prikazani na slikama 6.2 i 6.3. Rezultati detekcije čoveka u 5 frejmova na osnovu videa snimljenog na otvorenom prostoru je prikazan na slici 6.2. Slika 6.3 prikazuje rezultate detekcije dobijena na osnovu videa snimljenog u zatvorenom prostoru.



Slika 6.2. Detekcija čoveka na uzorku od nekoliko frejmova na videu snimljenom u otvorenom prostoru [19]



Slika 6.3. Detekcija čoveka na uzorku od nekoliko frejmova na videu snimljenom u zatvorenom prostoru

Koordinate čoveka dobijene sistemom za detekciju su kasnije iskorišćene u modulu za praćenje čija je osnova linearni Kalmanov filter koji rekurzivno predviđa i procenjuje poziciju čoveka u horizontalnoj ravni u cilju efikasnijeg upravljanja robotom.

Radi akvizicije slike, kamera Bumblebee XB3 je postavljena na najvišu tačku na robotu i sastoji se od tri kamere u liniji sa razmakom od 120 mm između centralne kamere i ostale dve. Svaka kamera poseduje vidno polje od 70° i rezoluciju od 1280×960 piksela. Prilikom eksperimenata korišćena je leva i desna spoljna kamera zbog veće dužine bazne linije između kamera, jer veća bazna linija daje preciznija izračunavanja rastojanja objekata koji su udaljeniji od kamere, te odatle bazna linija iznosi 240 mm.

U cilju procene performansi razvijenog sistema vizije, eksperimentalni rezultati su poređeni sa tačnim podacima koji su dobijeni na dva načina. U prvom eksperimentu (Slika 6.3) stvarna udaljenost čoveka je dobijena digitalnim laserom Bosch PLR 50 digital laser rangefinder [112]. Prema tehničkim karakteristikama, tačnost ovog uređaja je ± 2 mm, bez obzira na udaljenost od mete-čoveka. Modul stereo vizije je testiran na 1055 frejma i srednja vrednost greške distance je 2.38 %, dok je standardna devijacija iznosila 2.12 %.

U drugom eksperimentu čovek je pratio referentnu putanju koja je nacrtana na podu. U predloženom sistemu estimirana je pozicija čoveka na osnovu poznatog prethodnog stanja. Takođe je pretpostavljeno da je brzina čoveka između dva susedna frejma konstantna.

6.1.1. Inicijalizacija Kalmanovog filtera

Kao što je već ranije rečeno, pristup Kalmanovim filterom podrazumeva da se ponašanje objekta koji se kreće može opisati unapred definisanim modelima kretanja i merenja.

$$\mathbf{x}_k = \mathbf{F}_k \mathbf{x}_{k-1} + \mathbf{w}_k \quad (6.1)$$

$$\mathbf{y}_k = \mathbf{H}_k \mathbf{x}_k + \mathbf{v}_k \quad (6.2)$$

Gde je model kretanja čoveka predstavljen vektorom stanja:

$$\mathbf{x}_k = [x_{HC} \quad z_{HC} \quad \dot{x}_{HC} \quad \dot{z}_{HC}]^T \quad (6.3)$$

Prema tome, prenosna matrica stanja \mathbf{F}_k , koja određuje vezu između trenutnog stanja \mathbf{x}_k i prethodnog stanja \mathbf{x}_{k-1} , jednaka je:

$$\mathbf{F}_k = \begin{bmatrix} 1 & 0 & \Delta t & 0 \\ 0 & 1 & 0 & \Delta t \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (6.4)$$

Budući da se pozicija čoveka prati iz frejma u frejm, $\Delta t = 1$.

Prenosna matrica stanja \mathbf{F}_k postaje:

$$\mathbf{F}_k = \begin{bmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (6.5)$$

Vektori \mathbf{w}_k i \mathbf{v}_k predstavljaju beli šum koji su nezavisni jedan od drugog sa matricama kovarijansi $\mathbf{Q}_k = \mathbf{E}[\mathbf{w}_k \mathbf{w}_k^T]$ i $\mathbf{R}_k = \mathbf{E}[\mathbf{v}_k \mathbf{v}_k^T]$.

Matrica \mathbf{Q}_k meri varijabilnost ulaznog signala u donosu na “idealni” prelaz definisan matricom prelaza \mathbf{F}_k . Veće vrednosti u ovoj matrici znače da ulazni signal ima veću varijansu i da filter treba da bude prilagodljiviji. Manje vrednosti rezultiraju finijim izlaznim rezultatima, ali je filter manje adaptivan kod većih promena.

Kod linearnog Kalmanovog filtera, matrica šuma procesa je definisana kao:

$$\mathbf{Q}_k = \begin{bmatrix} \frac{\Delta t^4}{4} & 0 & 0 & 0 \\ 0 & \frac{\Delta t^4}{4} & 0 & 0 \\ 0 & 0 & \frac{\Delta t^4}{4} & 0 \\ 0 & 0 & 0 & \frac{\Delta t^4}{4} \end{bmatrix} \times \sigma_u^2 \quad (6.6)$$

gde je $\Delta t = 1$ i $\sigma_u^2 = 10^1$.

$$\mathbf{Q}_k = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \times 2.5 * 10^1 \quad (6.7)$$

Veza između vektora stanja \mathbf{x}_k i vektora merenja \mathbf{y}_k je definisana matricom \mathbf{H}_k . Polazeći od činjenice da je vektor stanja sačinjen od četiri elemenata, dok se vektor merenja sastoji od dva elementa, matrica \mathbf{H}_k mora biti veličine 4 puta 2.

$$\mathbf{H}_k = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} \quad (6.8)$$

Matrica \mathbf{R}_k definiše grešku mernog uređaja. Uobičajeno je da se tačnost mernog uređaja definiše od strane proizvođača, međutim kod sistema stereo vizije ovu tačnost se određuje empirijski.

Matrica šuma merenja, za linearni sistem, definisana je kao:

$$\mathbf{R}_k = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \times \sigma_v^2 \quad (6.9)$$

gde je predpostavljeno da je $\sigma_v^2 = 50$, te \mathbf{R}_k postaje:

$$\mathbf{R}_k = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \times 50 \quad (6.10)$$

Veće vredosti u ovoj matrici znače da se manja težina pripisuje samim merenjima, a veća modelu kretanja, dok manje vrednosti označavaju da se merenja smatraju tačnijim i samim tim veća težina se pripisuje samim merenjima a manja modelu.

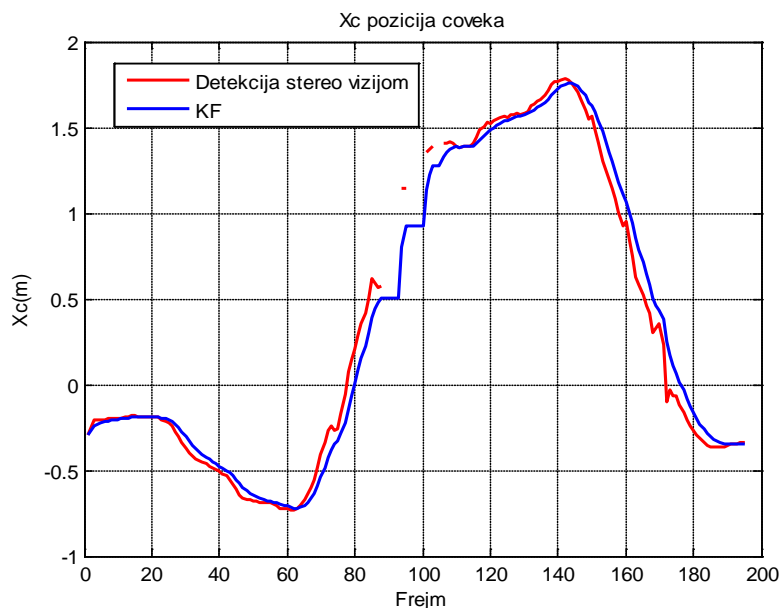
Posteriorna kovarijaciona matrica greške estimacije \mathbf{P}_k je mera tačnosti estimirane vrednosti stanja \mathbf{x}_k u trenutku k . Tokom vremena ova matrica se podešava samim filterom, ali je za inicijalizaciju filtera potrebna njena inicijalna vrednost \mathbf{P}_0 . Ukoliko su početna stanja poznata, tada se inicijalna kovarijaciona matrica može smatrati nula matricom. Inače, inicijalizuje se kao dijagonalna matrica. U tom slučaju filter će u nekoliko prvih koraka više oslanjati na informacije dobijene merenjem, pre nego na sam model merenja.

Inicijalna kovarijaciona matrica u razmatranom radu \mathbf{P}_0 definisana je kao:

$$\mathbf{P}_0 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \times 10^{-2} \quad (6.11)$$

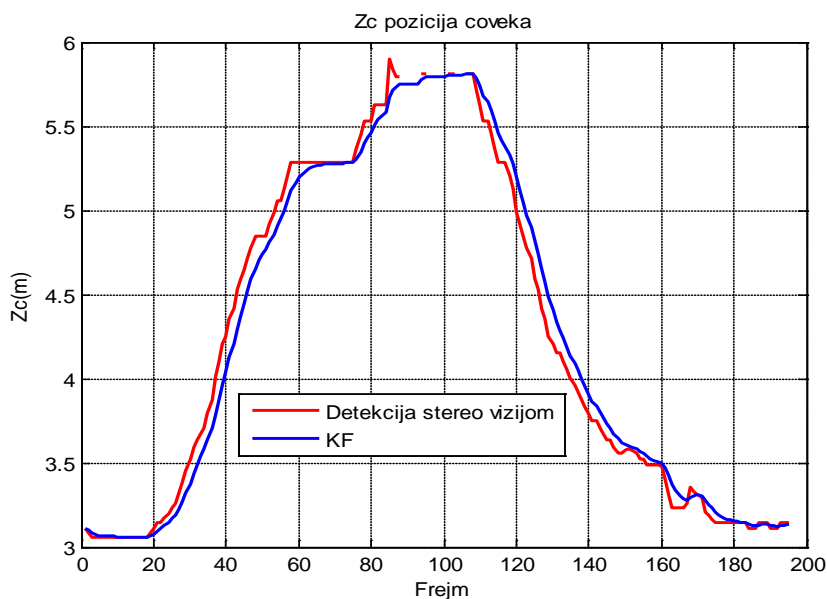
6.1.2. Rezultati praćenja čoveka na osnovu Kalmanovog filtera

Osobine i svojstva predstavljenog stereovizijskog sistema za praćenje čoveka zasnovanog na Kalmanovom filteru testirana su u zatvorenom prostoru. Parovi stereo slika su primani i obrađivani sa kamere u punoj rezoluciji od 12 frejmova u sekundi radi dobijanja položaja čoveka u prostoru u odnosu na koordinatni sistem kamere. Rezultati procene pozicije čoveka na osnovu stereo vizije koristeći Kalmanov filter su prikazani na slikama 6.4, 6.5 i 6.6.

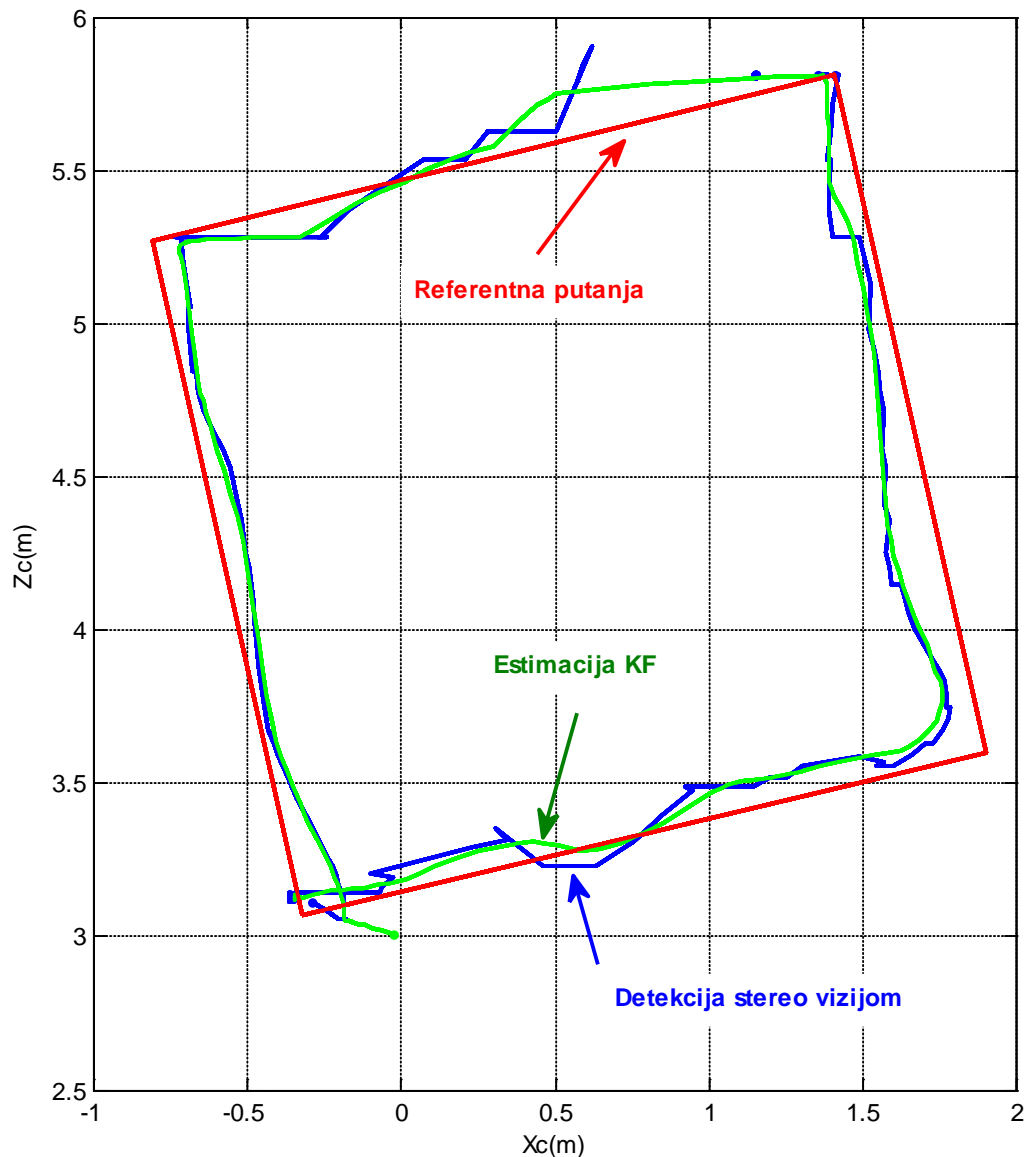


Slika 6.4. Prikaz rezultata dobijenih sistemom stereo vizije sa i bez modula praćenja (KF) u pravcu

X_C ose



Slika 6.5. Prikaz rezultata dobijenih sistemom stereo vizije sa i bez modula praćenja (KF) u pravcu Z_C ose



Slika 6.6. Praćenje referentne putanje kretanja čoveka predloženim stereovizijskim sistemom sa i bez modula praćenja zasnovanim na Kalmanovom filteru (KF)

Na dijagramima prikazanim na slikama 6.4 i 6.5 prikazane su promene koordinata položaja čoveka i upoređeni su rezultati dobijeni stereovizijskim sistemom sa i bez modula praćenja zasnovanim na Kalmanovom filteru. Slika 6.6 prikazuje pogled na scenu u X_cZ_c ravni i na njoj referentnu putanju kvadratnog oblika, 2.3×2.3 m, koja je predstavljena crvenom linijom, dok je izlaz iz predloženog sistema stereo vizije za detekciju položaja čoveka sa i bez modula za praćenje, predstavljen plavom i zelenom bojom respektivno.

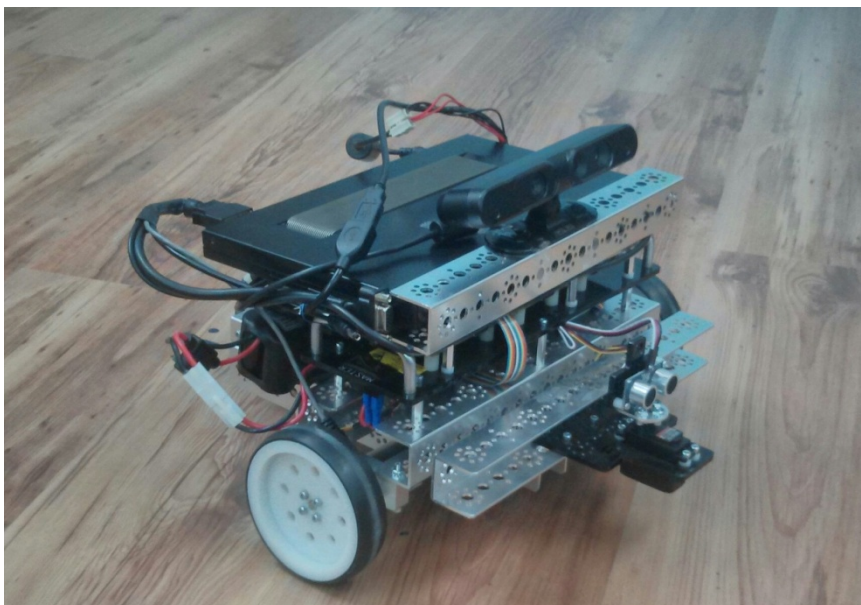
Na osnovu slika primećuje se da je Kalmanov filter sposoban da smanji šum u merenjima i da predvidi stanje sistema čak i kada sistem stereo vizije nije dao nikakva merenja u određenim intervalima vremena zbog neočekivanog poremećaja u sistemu. Tri takva vremenska intervala se

mogu videti na dijagramima gde se prekida crvena linija na slikama 6.4 i 6.5. Takođe se može primetiti i u gornjem desnom uglu kvadrata gde dolazi do prekida plave linije na slici 6.6.

Ovakva kontinualna estimacija položaja čoveka je veoma važno da bi se obezbedili ulazi za upravljački sistem robota koji treba da prati čoveka.

6.2. MOBILNA ROBOTSKA PLATFORMA DANI SA ASUS XTION 3D SENZOROM

Za potrebe ove teze, na Mašinskom fakultetu, razvijeni podmodul za praćenje čoveka testiran je na drugoj robotskoj platformi gde ASUS Xtion 3D senzor pruža vizualne informacije o okolini robota (Slika 6.7). Pored Kalmanovog filtera, testiran je Partikle filter za predviđanje i procenu pozicije čoveka, a izvršena je i njihova uporedna analiza za rešavanje problema praćenja ljudi.



Slika 6.7: DaNI robot opremljen industrijskim računarom i ASUS Xtion PRO LIVE kamerom

Pokretna robotska platforma NI Robotics Starter Kit 2.0, poznat kao DaNI robot je razvijen od strane kompanije National Instruments. Robot poseduje senzore, motore, i NI Single-Board RIO-9632 kontrolnu ploču sa procesorom za rad u realnom vremenu. Korišćeni mobilni robot poseduje dva točka koji se pogone sa dva motora i jedan pomoćni točak bez pogona.

Asus Xtion PRO LIVE 3D senzor (Slika 6.8) je povezan sa industrijskim računarom koji je postavljen na DaNI mobilnu robotsku platformu firme National Instruments, a koji preko Lan kabla komunicira sa robotom (Slika 6.7).

Asus Xtion PRO LIVE 3D senzor radi kao senzor strukturane svetlosti [121]. Senzor se sastoji od infracrvene laserske diode (IR) koja emituje svetlost po određenom obrascu i IR kamere za snimanje reflektovane svetlosti. Dubina se izračunava softverski na osnovu dispariteta reglektovanih obrazaca u odnosu na referentni obrazac. Dopunska RGB kamera je dodata da bi se obezbedile dodatne informacije o boji i teksturi površi.

Asus Xtion PRO LIVE 3D senzor ostvaruje softversku vezu sa računarnom preko OpenNI softvera („open source“okruženje za3D senzore). OpenNI omogućuje pristup PrimeSense komatibilanim 3D sensorima, omogućujući inicijalizaciju senzora i dobijanja dubine, RGB i infrared videa sa samog uređaja.

Simulink podrška za 3D senzor sastoji se od IMAQ, Image, IR, Depth, Motion i Skeletonblokova, a u cilju postizanja „real-time“ akviziciju podataka sa komercijalnih 3D senzora kaošto su Microsoft Kinect i Asus Xtion kamere.Sva softverska rešenja za obradu slike, klasifikaciju i detekciju realizovana su u softverskom paketu Simulink kompanije MathWorks. U tabeli 6.2 data je specifikacija Asus Xtion live 3D senzora.



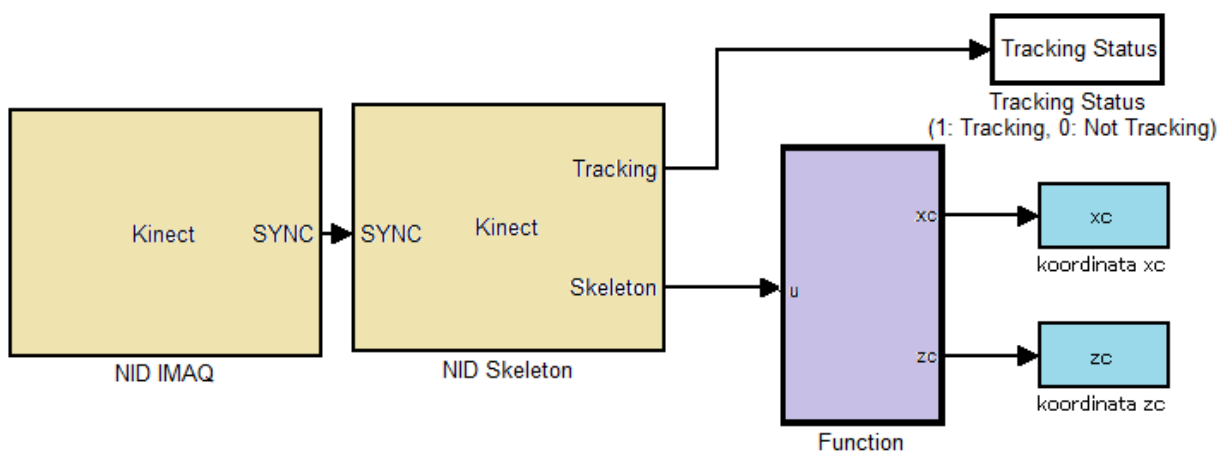
Slika 6.8: ASUS Xtion PRO LIVE 3D senzor [11]

Tabela 6.2: Specifikacije Asus Xtion live 3D senzora [11]

| | | | |
|-------------------------|---|-------------------|---|
| Senzor | RGB, Infrared senzor, 2*Mikrofon | Softver | OPEN NI SDK paket |
| Veličina dubinske slike | VGA (640x480) : 30 fps QVGA (320x240): 60 fps | Programski jezik | C++/C# (Windows) C++(Linux), JAVA |
| Rezolucija | SXGA (1280x1024) | Radno okruženje | Unutrašnjost |
| Vidno polje | 58° H, 45° V, 70° D (Horizontalno, Vertikalno, Dijagonalno) | Operativni sistem | Win 32/64 : XP, Vista, 7, 8, Linux Ubuntu 10.10: X86,32/64 bit, Android(by request) |

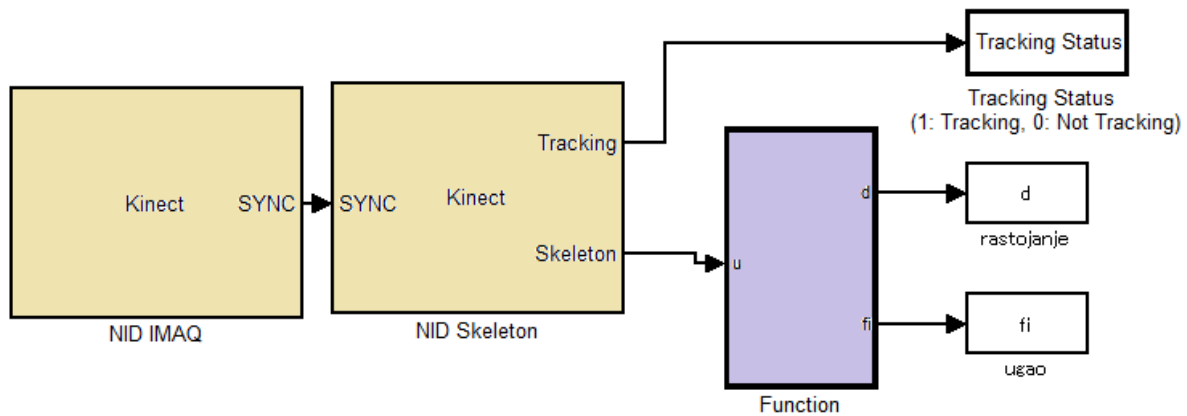
| | | | |
|--------------------------------------|-----------------|-----------|-----------------|
| Domet (preporučena udaljenost) | od 0.8m do 3.5m | Interfejs | USB2.0 |
| Dimenzije | 18 x 3.5 x 5 cm | Platforma | Intel X86 & AMD |

Kako bi se dobili valjani podaci za dalju obradu i konačno upravljanje mobilnom robotskom platformom, bilo je neophodno prvo razviti model za detekciju čoveka iz slika dobijenih senzorom robotske vizije. Na slici 6.9 je prikazan model za prikupljanje podataka dobijenih iz 3D senzora i detekciju pozicije centra čoveka, tj. njegovog torzoa, pri čemu za potrebe ove teze, dobijane su samo x_c , z_c koordinate čoveka u horizontalnoj ravni u koordinatnom sistemu kamere.

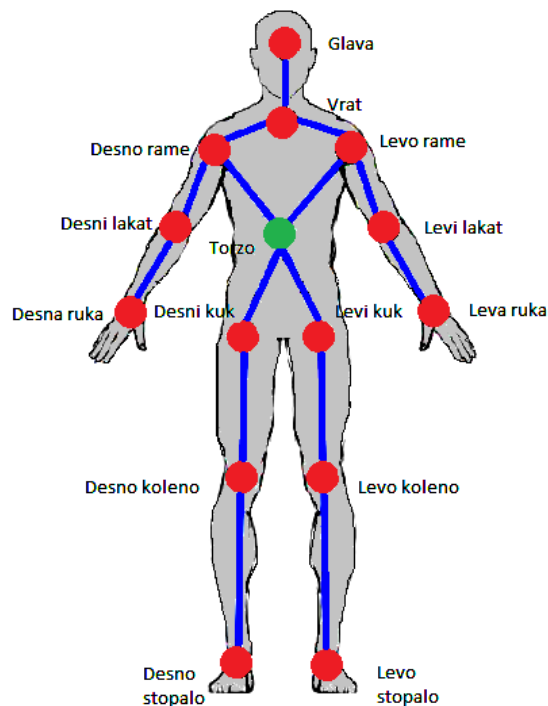


Slika 6.9: Model za prikupljanje podataka i detekciju x_c i z_c koordinate čoveka u horizontalnoj ravni u koordinatnom sistemu kamere

Na slici 6.10 je prikazan model za prikupljanje podataka dobijenih iz 3D senzora, pri čemu se dobijaju podaci o rastojanju d čoveka od koordinatnog početka kamere u horizontalnoj ravni, kao i ugao orijentacije φ koji daje informaciju o tome koliko je čovek pomeren u odnosu na Z_c osu u horizontalnoj ravni u koordinatnom sistemu kamere.



Slika 6.10: Model za detekciju čoveka i prikupljanje podataka o njegovom rastojanja d i orijentacije φ u horizontalnoj ravni u koordinatnom sistemu kamere

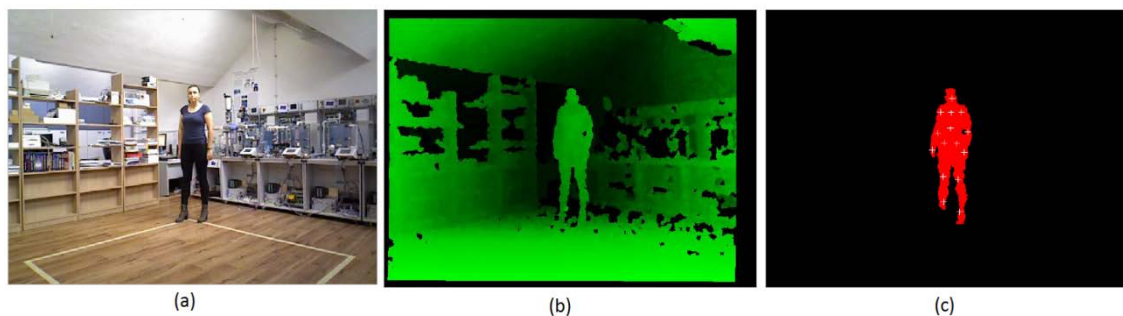


Slika 6.11. Tačke od interesa detektovane Skeleton blokom

IMAQ blok prikuplja podatke dobijene od kamere, dok se *Skeleton* blok koristi za nalaženje koordinata tačaka od interesa (koordinate glave čoveka, torzo, itd.). U bloku *function* biraju se tačke od interesa koje je potrebno pratiti koje su predstavljene brojevima. U predstavljenom radu, to je centar čoveka, tj. njegov/njen torzo.

Tačke od interesa koje se mogu dobiti iz *Skeleton* bloka kod OpenNi okruženja su prikazane na slici 6.11.

Na slici 6.12 je prikazan primer originalne, dubinske i slike sa detektovanim čovekom i tačkama od interesa dobijenih ASUS Xtion PRO LIVE 3D senzorom.



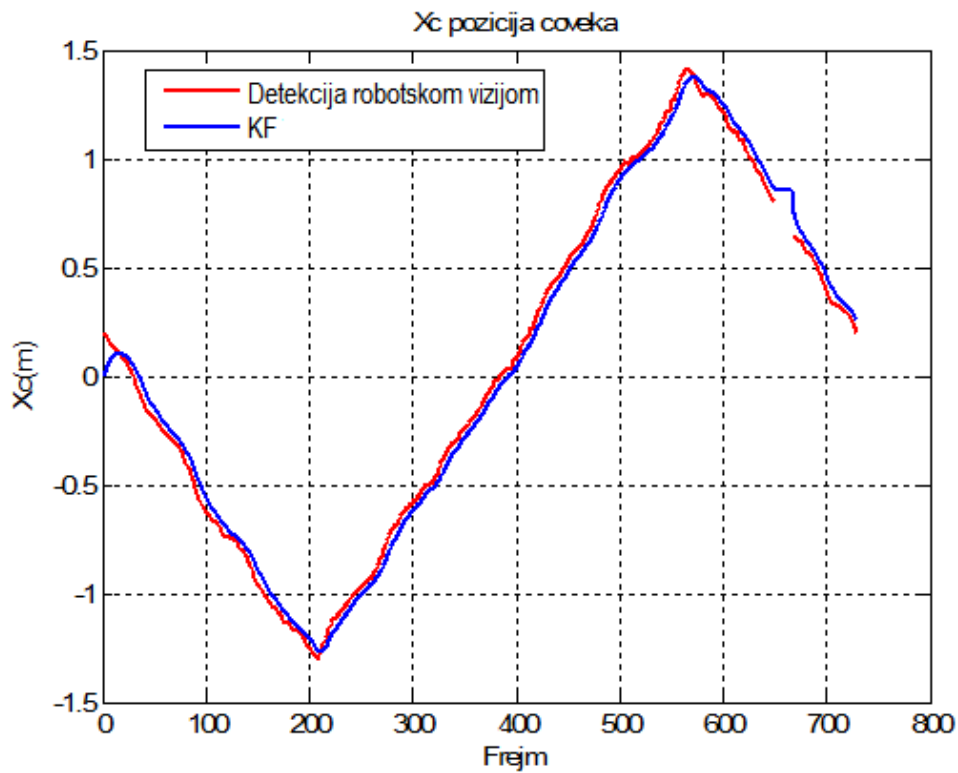
Slika 6.12:a) Originalna slika, b) Dubinska slika (*eng.* Depth image), c) Slika sa detektovanim čovekom i tačkama od interesa

U eksperimentima izvođenim sa ovom robotskom platformom, robot je prvo samo posmatrao osobu bez da je prati pri čemu je čovek pratio referentnu putanju koja je nacrtana na podu (Slika 6.12.a), a zatim su rađeni eksperimenti gde robot prati osobu u realnom vremenu o čemu će biti reči u narednom poglavlju.

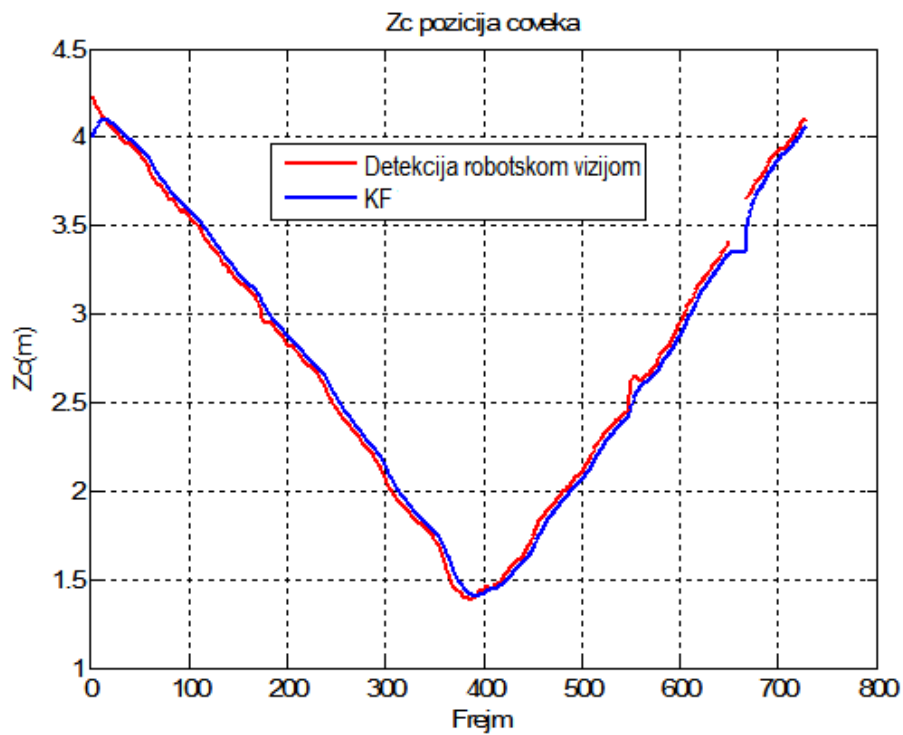
6.2.1 Rezultati praćenja čoveka na osnovu Kalmanovog filtera

U ovom eksperimentu parovi stereo slika su primani i obrađivani sa kamere u rezoluciji od 30 frejmova u sekundi radi dobijanja položaja čoveka u horizontalnoj ravni u odnosu na koordinatni sistem kamere. Podaci dobijeni sa kamere su predstavljeni vektorom merenja $y_k = [x_{HC} z_{HC}]^T$, što znači da je problem posmatran kao linearni problem. Inicijalizacija Kalmanovog filtera je slična kao i u poglavlju 6.1.1., pri čemu je pretpostavljeno da je varijansa matrice šuma Q_k procesa jednaka $\sigma_u^2 = 1$, a varijansa matrice merenja R_k jednaka je $\sigma_v^2 = 10^2$. Inicijalna kovarijaciona matrica P_0 je ista kao i u poglavlju 6.1.1.

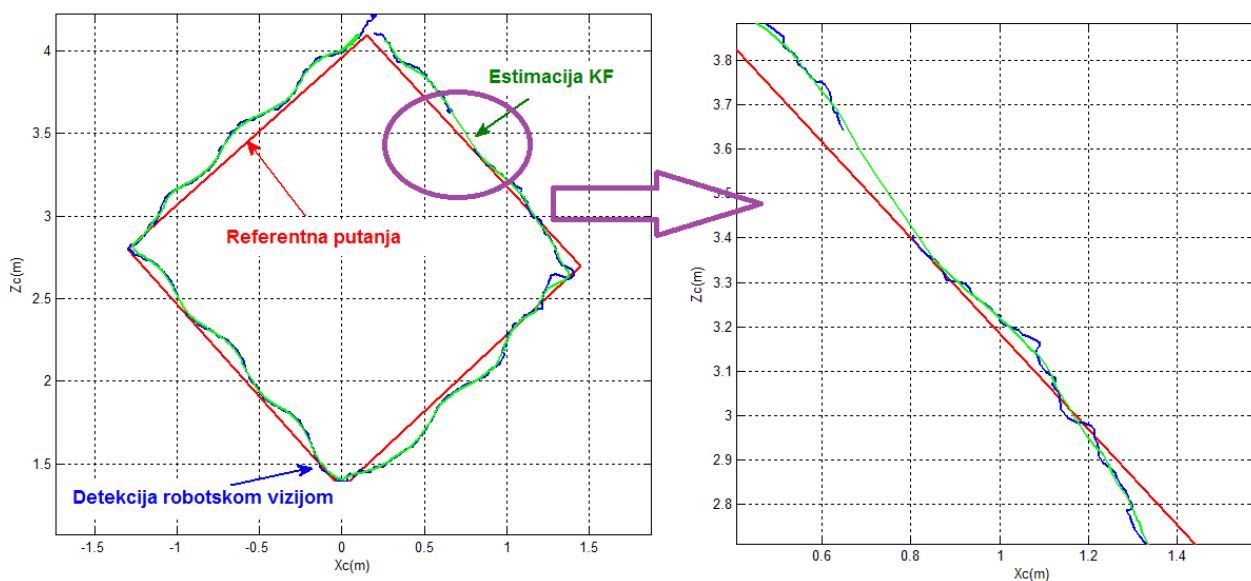
Rezultati procene pozicije čoveka na osnovu podataka dobijenih ASUS Xtion PRO LIVE 3D senzorom koristeći Kalmanov filter su prikazani na slikama 6.13, 6.14 i 6.15.



Slika 6.13. Prikaz rezultata dobijenih ASUS Xtion PRO LIVE 3D senzorom sa i bez modula praćenja (KF) u pravcu X_C ose



Slika 6.14. Prikaz rezultata dobijenih ASUS Xtion PRO LIVE 3D senzorom sa i bez modula praćenja (KF) u pravcu Z_C ose



Slika 6.15. Praćenje referentne putanje ASUS Xtion PRO LIVE 3D senzorom sa i bez modula praćenja zasnovanim na Kalmanovom filteru

Na dijagramima prikazanim na slici 6.13 i 6.14 prikazane su promene koordinata položaja čoveka i upoređeni su rezultati dobijeni ASUS Xtion PRO LIVE 3D senzorom sa i bez modula praćenja zasnovanim na Kalmanovom filteru.

Slika 6.15 prikazuje pogled na scenu u X_cZ_c ravni i na njoj referentnu putanju kvadratnog oblika, 2×2 m, koja je predstavljena crvenom linijom, dok je izlaz iz sistema stereo vizije za detekciju položaja čoveka sa ASUS Xtion PRO LIVE 3D senzorom sa i bez modula praćenja, predstavljen plavom i zelenom bojom respektivno. Na slikama se može primetiti da u jednom vremenskom intervalu nema merenja, ali da modul praćenja uspešno procenjuje poziciju čoveka. Na slici 6.15 deo gde nema merenja je uvećan, zbog jasnijeg prikaza rezultata.

6.2.2. Rezultati praćenja čoveka na osnovu proširenog Kalmanovog filtera

Kao što je već rečeno, modul praćenja treba da bude primenljiv kod različitih robotskih platformi i kod različitih senzora detekcije položaja čoveka. Prema tome, za razvoj modula praćenja za slučajeve kada je sistem nelinearan, tj. za rezultate merenja dobijaju se rastojanje d i ugao orijentacije φ , razvijeni su algoritmi praćenja na osnovu proširenog Kalmanovog filtera i Partikle filtera. Za izvođenje ovog eksperimenta parovi stereo slika su primani i obrađivani sa kamere u rezoluciji od 30 frejmova u sekundi radi dobijanja položaja čoveka u horizontalnoj ravni u odnosu na koordinatni sistem kamere. Korišćen je model za prikupljanje dobijenih podataka prikazan na slici 6.10.

U ovom eksperimentu podrazumeva se da se ponašanje objekta koji se kreće, čoveka, može opisati unapred definisanim linearnim modelom kretanja i nelinearnim modelom merenja.

$$\mathbf{x}_k = \mathbf{F}_k \mathbf{x}_{k-1} + \mathbf{w}_k \quad (6.12)$$

$$\mathbf{y}_k = h_k(\mathbf{x}_k, \mathbf{v}_k) \quad (6.13)$$

gde je vektor stanja

$$\mathbf{x}_k = [x_{HC} \quad z_{HC} \quad \dot{x}_{HC} \quad \dot{z}_{HC}]^T \quad (6.14)$$

A vektor merenja

$$\mathbf{y}_k = \begin{pmatrix} d \\ \varphi \end{pmatrix} = \begin{pmatrix} \sqrt{x_{HC}^2 + z_{HC}^2} \\ \arctg \frac{x_{HC}}{z_{HC}} \end{pmatrix} \quad (6.15)$$

Prema tome, prenosna matrica stanja \mathbf{F}_k , koja određuje vezu između trenutnog stanja \mathbf{x}_k i predhodnog stanja \mathbf{x}_{k-1} , jednaka je:

$$\mathbf{F}_k = \begin{bmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (6.16)$$

Veza između vektora stanja \mathbf{x}_k i vektora merenja \mathbf{y}_k je definisana matricom \mathbf{H}_k , koja predstavlja Jakobianovu matricu sa parcijalnim izvodima sistemske funkcije h u odnosu na stanje \mathbf{x}_k , $\mathbf{H}_k = \left. \frac{\partial h}{\partial \mathbf{x}} \right|_{(\tilde{\mathbf{x}}_k, 0)}$.

$$\mathbf{H}_k = \left. \frac{\partial h}{\partial \mathbf{x}} \right|_{(\tilde{\mathbf{x}}_k, 0)} = \begin{bmatrix} \frac{x_{HC}}{\sqrt{x_{HC}^2 + z_{HC}^2}} & \frac{z_{HC}}{\sqrt{x_{HC}^2 + z_{HC}^2}} & 0 & 0 \\ \frac{z_{HC}}{x_{HC}^2 + z_{HC}^2} & \frac{x_{HC}}{x_{HC}^2 + z_{HC}^2} & 0 & 0 \end{bmatrix} \quad (6.17)$$

Za šumove \mathbf{w}_k i \mathbf{v}_k smo je pretpostavljeno da su nezavisni i nepromenljivi u toku vremena, pri čemu je pretpostavljeno da je varijansa matrice šuma \mathbf{Q}_k procesa jednaka $\sigma_u^2 = 1$, a varijansa matrice merenja \mathbf{R}_k jednaka je $\sigma_v^2 = 10^2$.

Matrica šuma procesa je definisana kao:

$$\mathbf{Q}_k = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \times \sigma_u^2 \quad (6.18)$$

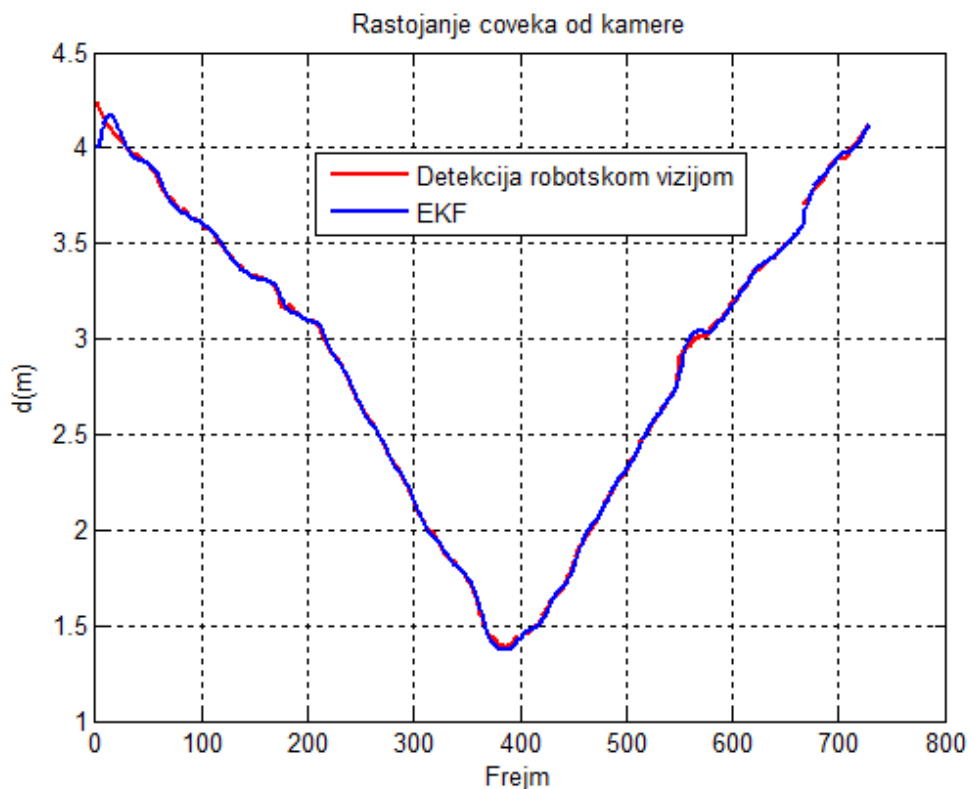
Matrica šuma merenja je definisana kao:

$$\mathbf{R}_k = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \times \sigma_v^2 \quad (6.19)$$

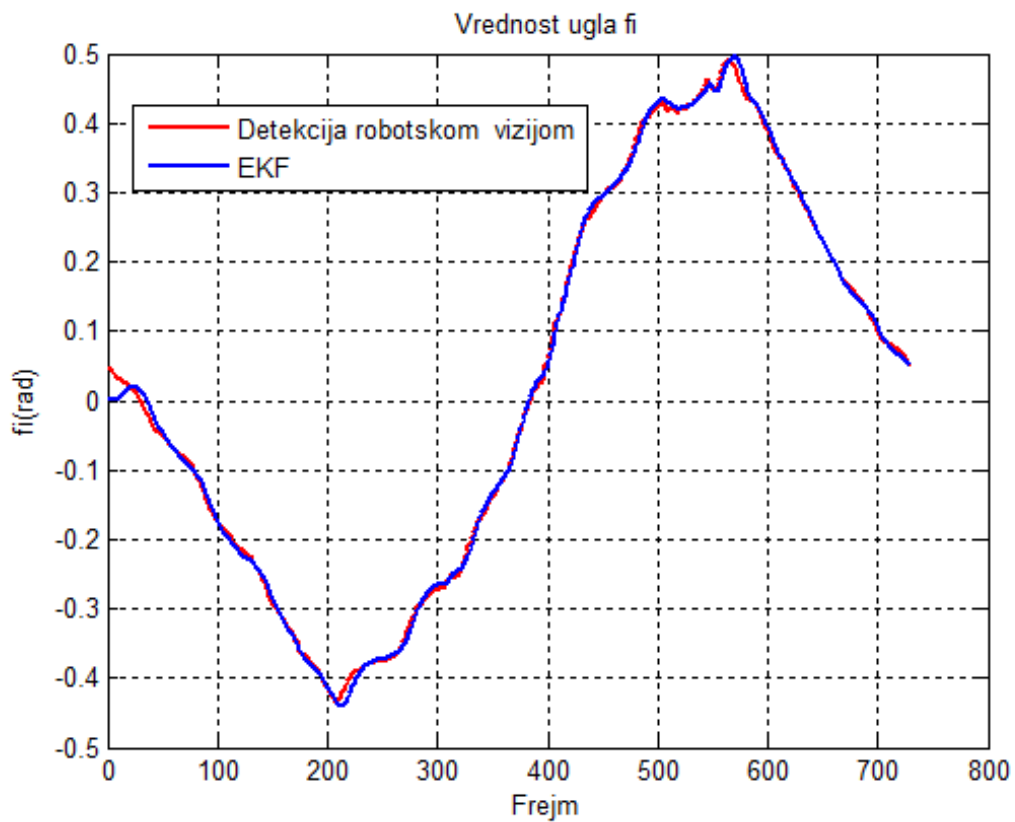
Inicijalna kovarijaciona matrica \mathbf{P}_0 definisana je kao

$$\mathbf{P}_0 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \times 10^{-2} \quad (6.20)$$

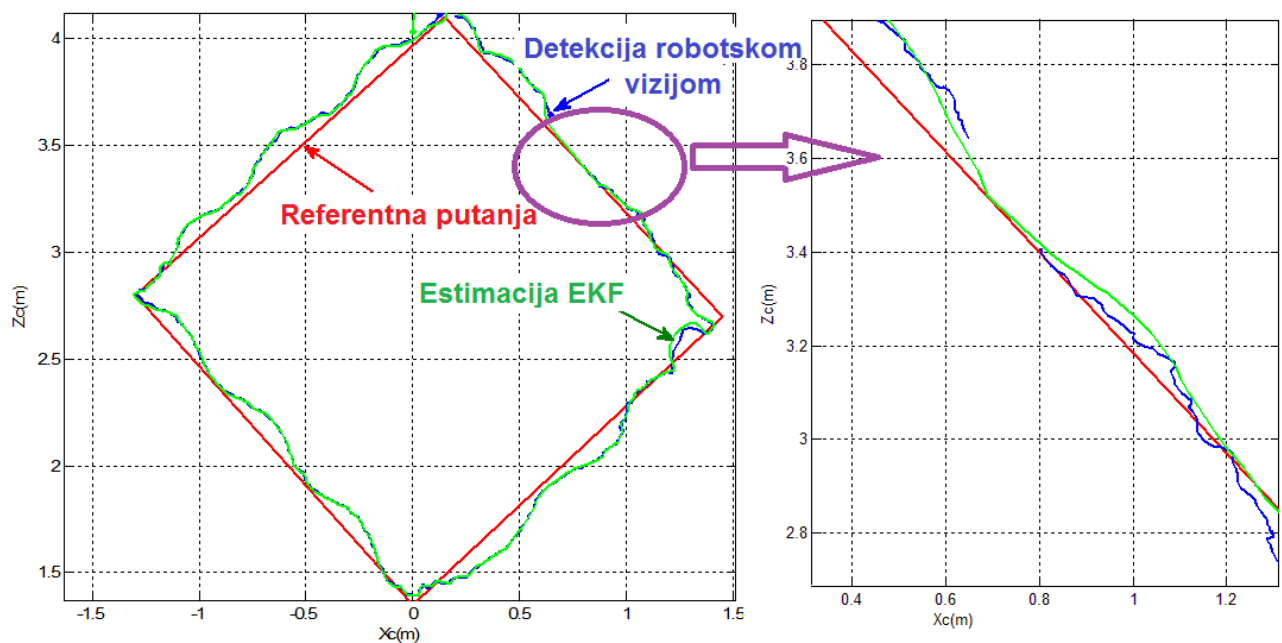
Rezultati procene pozicije čoveka na osnovu podataka dobijenih ASUS Xtion PRO LIVE 3D senzorom koristeći prošireni Kalmanov filter su prikazani na slikama 6.16, 6.17 i 6.18. Zbog velikog broja podataka, na slici 6.18. uvećan je deo dijagrama da bi se jasnije videli rezultati procene proširenim Kalmanovim filterom.



Slika 6.16. Prikaz rezultata dobijenih ASUS Xtion PRO LIVE 3D senzorom sa i bez modula – rastojanje d



Slika 6.17. Prikaz rezultata dobijenih ASUS Xtion PRO LIVE 3D senzorom sa i bez modula –ugao orijentacije φ



Slika 6.18. Praćenje referentne putanje ASUS Xtion PRO LIVE 3D senzorom sa i bez modula praćenja zasnovanim na proširenom Kalmanovom filteru

6.2.3. Rezultati praćenja čoveka na osnovu Partikle filtera

Postupak estimacije položaja čoveka kada se ulazni podaci u estimator tj. izlaz iz modula detekcije dobija u vidu rastojanje d i ugao orijentacije φ , testiran je i pomoću Partikle filtera sa 2000 čestica. Preporuka u literaturi je koristiti više od 1000 za dobijanje zadovoljavajućih rezultata, a minimalno je 500 čestica [120].

Modeli procesa i opservacije i -te partikle su dati izrazom:

$$\tilde{\mathbf{x}}_k^{(i)} = \mathbf{F}_k \mathbf{x}_{k-1}^{(i)} + \mathbf{Q}_k \cdot rand(0,1) \quad (6.21)$$

$$\tilde{\mathbf{y}}_k^{(i)} = h_k(\mathbf{x}_k^{(i)}, \mathbf{R}_k \cdot rand(0,1)) \quad (6.22)$$

gde je $\tilde{\mathbf{x}}_k^{(i)}$ predikcija vektora stanja, a $\tilde{\mathbf{y}}_k^{(i)}$ predikcija merenja u trenutku k i -tog partikla.

Matrica prelaza stanja je \mathbf{F}_k je ista kao i kod proširenog Kalmanovog filtera, matrica šuma procesa \mathbf{Q}_k ima varijansu $\sigma_u^2 = 10^{-2}$, dok je vektor transformacije predikcije h , jednak

$$\mathbf{h}_k = \begin{pmatrix} \sqrt{x_{HC}^{(i)2} + z_{HC}^{(i)2}} \\ arctg \frac{x_{HC}^{(i)}}{z_{HC}^{(i)}} \end{pmatrix} \quad (6.23)$$

Težinski koeficijenti se izračunavaju na osnovu funkcije verovatnoće normalne raspodele

$$\xi_k^{(i)} = \frac{1}{2\pi\sqrt{\det(R)}} \cdot e^{-\frac{(d^T R^{-1} d)}{2}} \quad (6.24)$$

Dobijene težinske koeficijente normiramo

$$\tilde{\xi}^{(i)} = \frac{\xi^{(i)}}{\sum_{j=1}^N \xi^{(j)}} \quad (6.25)$$

Zatim sledi procena stanja u skladu sa sledećim izrazom

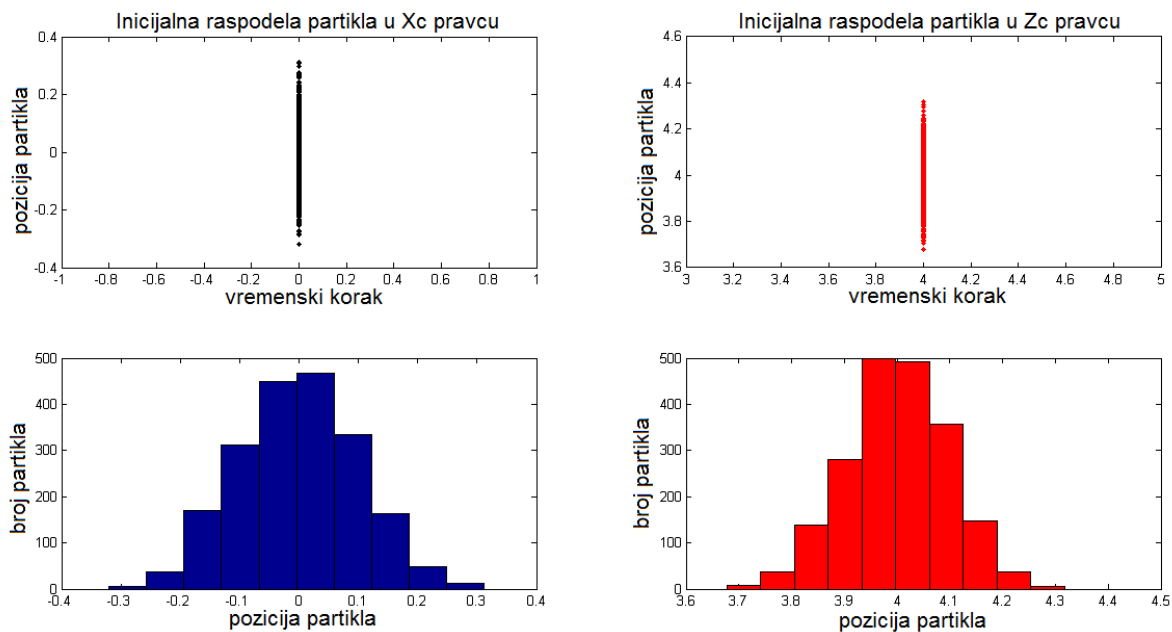
$$\hat{\mathbf{x}}_k^{(i)} = \sum_{i=1}^N \tilde{\xi}^{(i)} \cdot \tilde{\mathbf{x}}_k^{(i)} \quad (6.26)$$

Matrica šuma merenja je definisana kao:

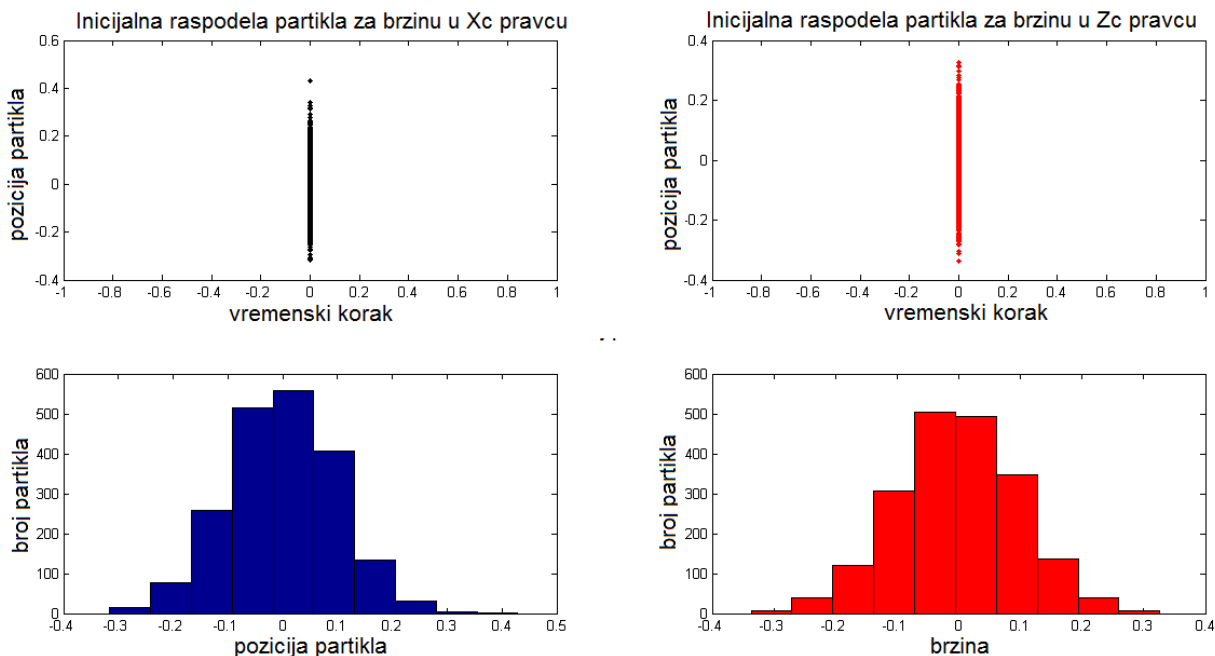
$$\mathbf{R}_k = \begin{bmatrix} 1 & 0 \\ 0 & 0.1 \end{bmatrix} \times \sigma_v^2 \quad (6.27)$$

gde je $\sigma_v^2 = 10^{-2}$.

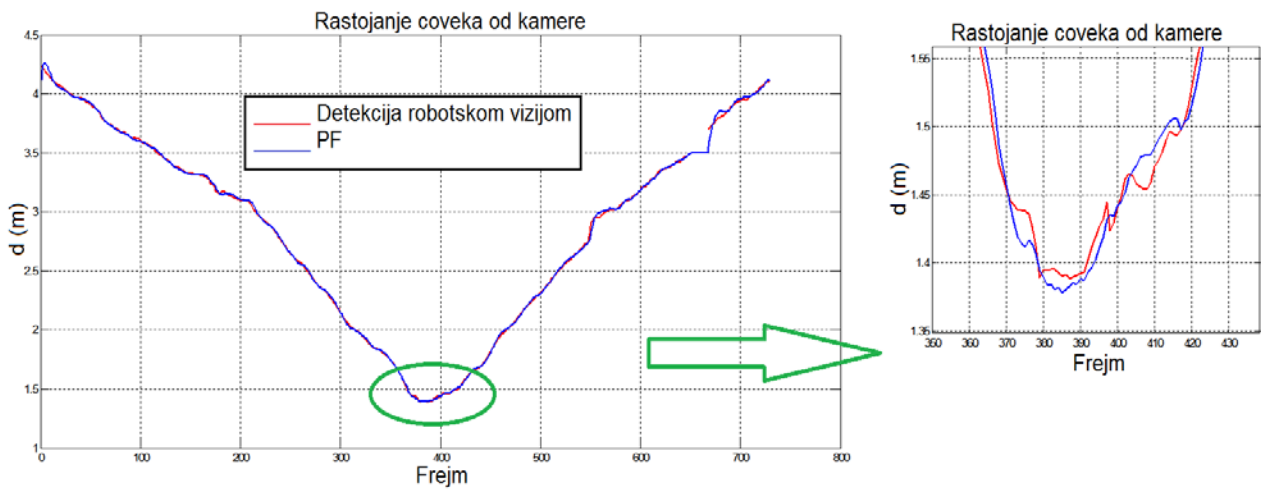
Rezultati procene pozicije čoveka na osnovu podataka dobijenih ASUS Xtion PRO LIVE 3D senzorom koristeći Partikle filter, kao i inicijalna raspodela partikla su prikazani na slikama 6.19, 6.20, 6.21, 6.22 i 6.23. Zbog velikog broja podataka, na slikama 6.21, 6.22 i 6.23 je uvećan deo dijagrama da bi se jasnije videli rezultati procene Partikle filterom.



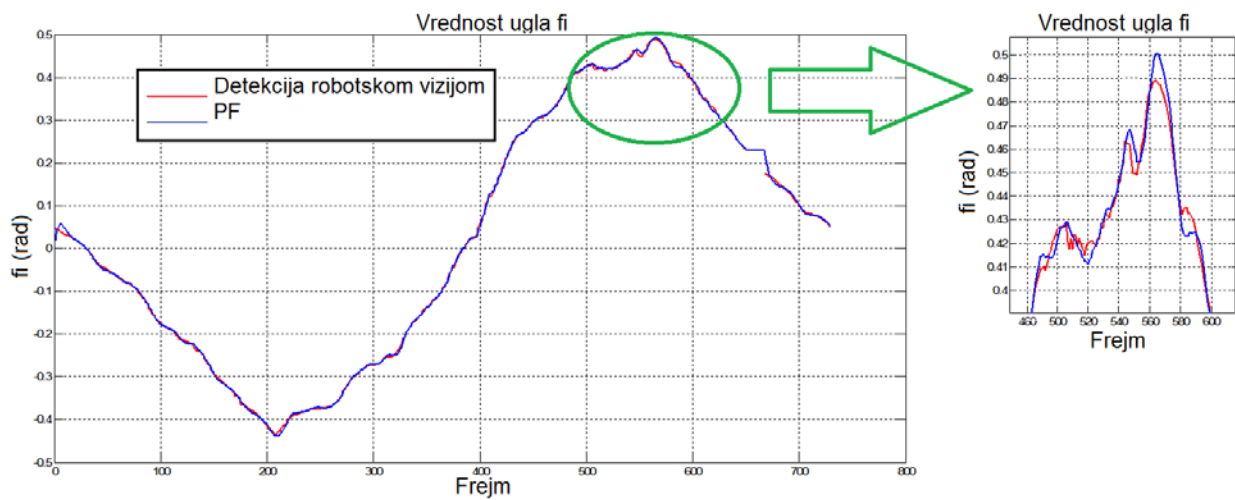
Slika 6.19. Inicijalna raspodela partikla za procenu pozicije u pravcu x_c i z_c ose



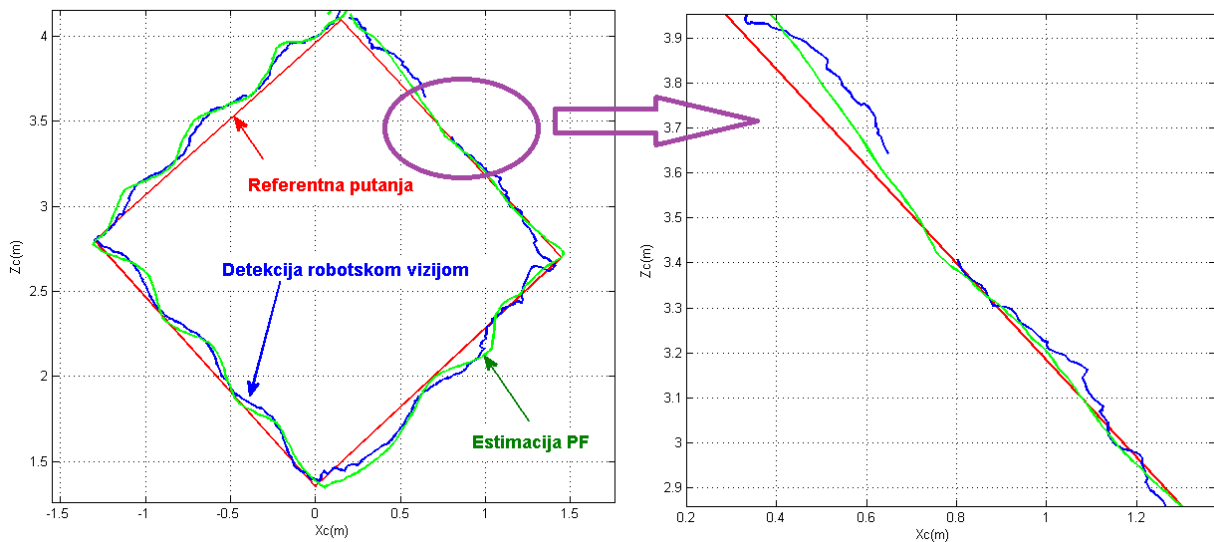
Slika 6.20. Inicijalna raspodela partikla za procenu brzina u pravcu x_c i z_c ose



Slika 6.21. Prikaz rezultata dobijenih ASUS Xtion PRO LIVE 3D senzorom sa i bez modula Partikle filterom – rastojanje d



Slika 6.22. Prikaz rezultata dobijenih ASUS Xtion PRO LIVE 3D senzorom sa i bez modula sa Partikle filterom –ugao orijentacije φ



Slika 6.23. Praćenje referentne putanje ASUS Xtion PRO LIVE 3D senzorom sa i bez modula praćenja zasnovanim na Partikle filteru

6.3. UPOREDNA ANALIZA ESTIMATORA

Uparedna analiza izvedena je tako što je merena greška estimacije, pri čemu se čovek kretao po pravoj liniji dužine 2 m (Slika 6.24).

Uparedna analiza je sprovedena na osnovu MSE i RMSE kriterijuma, koji se izračunavaju prema sledećem izrazu:

$$MSE = \frac{1}{M} \sum_{k=1}^M e_k^2 \quad (6.28)$$

$$RMSE = \sqrt{\frac{1}{M} \sum_{i=1}^N e_k^2} \quad (6.29)$$

I standardne devijacije SD, koja se računa po sledećoj formuli:

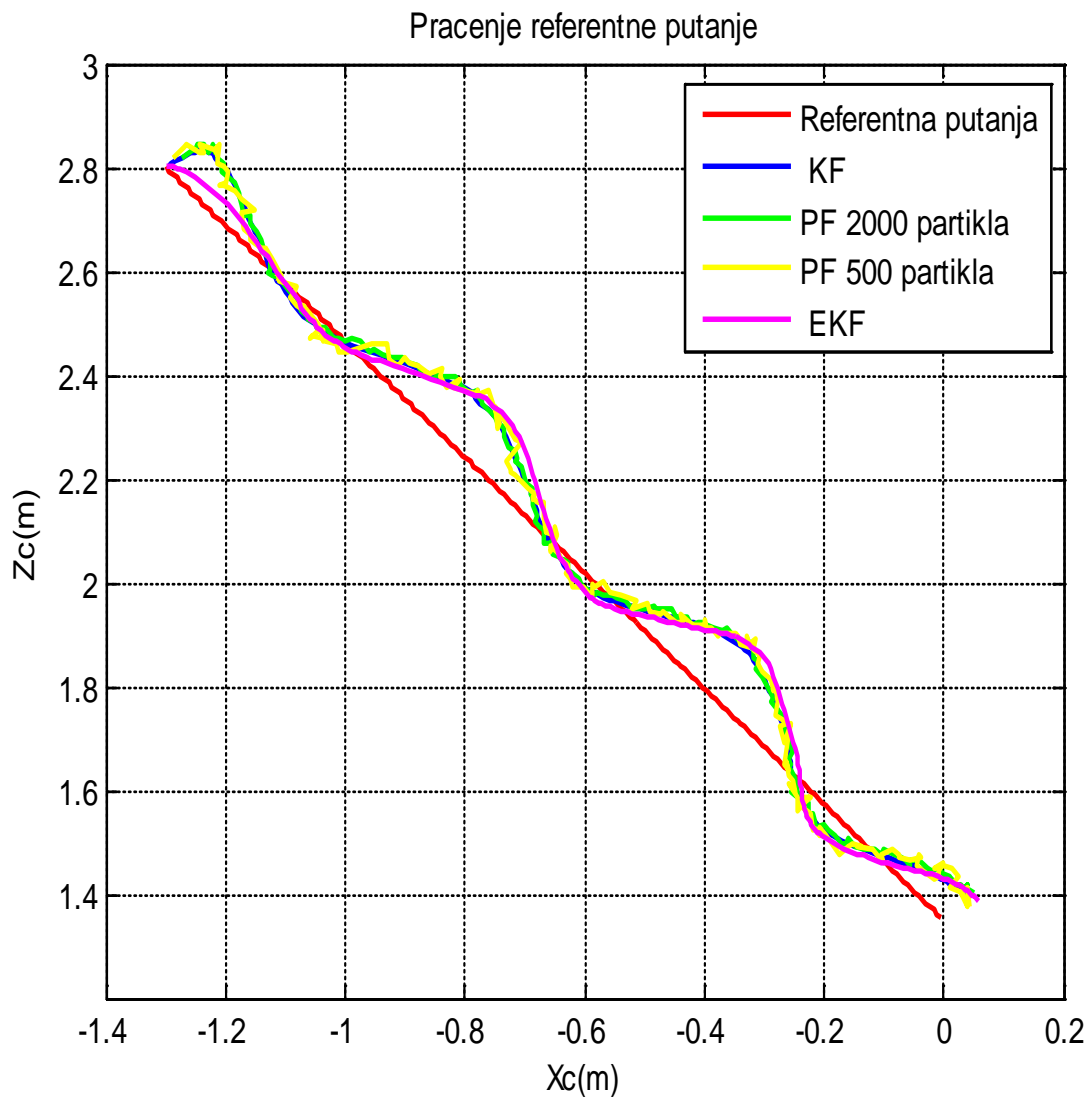
$$SD = \sqrt{\frac{1}{M} \sum_{i=1}^N (e_k - \bar{e})^2} \quad (6.30)$$

$$\bar{e} = \frac{1}{M} \sum_{i=1}^M e^2 \quad (6.31)$$

Gde je e_k greška pozicije definisana kao:

$$e_k = \sqrt{(\hat{x}_{HC_k} - x_{HC_k}^*)^2 + (\hat{z}_{HC_k} - z_{HC_k}^*)^2} \quad (6.32)$$

\hat{x}_{HC_k} i \hat{z}_{HC_k} predstavljaju estimirane koordinate u pravcu x_c i z_c , u koordinatnom sistemu kamere dobijene filterom $x_{HC_k}^*$ i $z_{HC_k}^*$ su tačne pozicije čoveka koji se kreće po pravoj liniji iscrtanoj na podu dužine 2 m. M je broj frejmova snimljenih tokom izvođenja ovog eksperimenta.

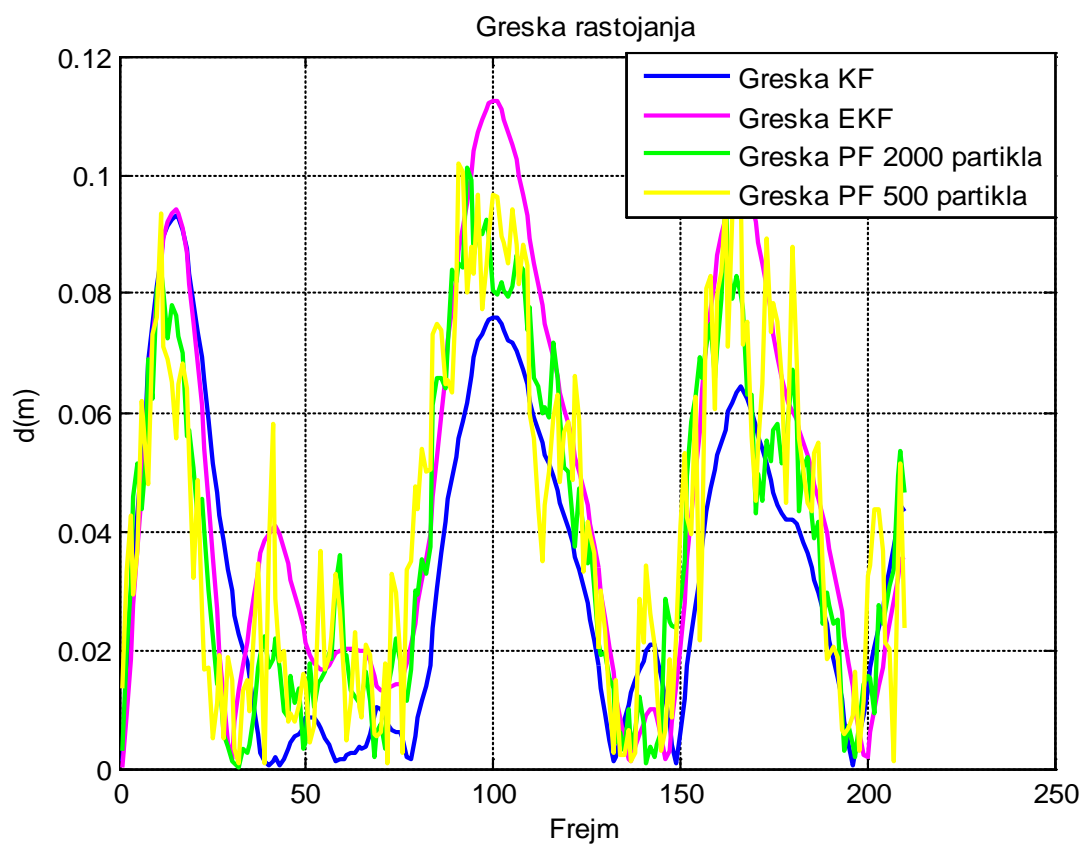


Slika 5.24. Praćenje referentne putanje modulom za praćenje zasnovanim na Kalmanovom filteru (plava), proširenim Kalmanovom filteru (ljubičasta), Partikle filteru sa 2000 čestica (zeleni) i Partikle filteru sa 500 čestica (žuta)

U tabeli 6.3 dati su rezultati ovog eksperimenta, dok je na slici 6.25 dat grafički prikaz apsolutne vrednosti grešaka tokom vremena.

Tabela 6.3: Greške praćenja

| | KF | EKF | PF 2000 | PF 500 |
|----------|--------|--------|---------|--------|
| MSE (m) | 0.0018 | 0.0030 | 0.0024 | 0.0026 |
| RMSE (m) | 0.0424 | 0.0529 | 0.0485 | 0.0506 |
| SD (m) | 0.0378 | 0.0442 | 0.0401 | 0.0443 |



Slika 6.25. Apsolutna greška praćenja rastojanja

Na osnovu dobijenih rezultata može se zaključiti da linearni Kalmanov filter ima najbolje karakteristike. U slučajevima kada je korišćen model sistem bio nelinearan, Partikle filter daje bolje rezultate. Naravno što je više čestica korišćeno, rezultat je bolji.

Međutim, mana Partikle filtera je što ne bi mogao da radi u realnom vremenu budući da vreme koje je potrebno za dobijanje estimacije pozicije Partikle filterom sa 2000 čestica iznosi oko 0.738 s, sa 500 čestica je 0.1325 s dok je Kalmanovim filterom 0.0142 s, a proširenim Kalmanovim 0.018 s.

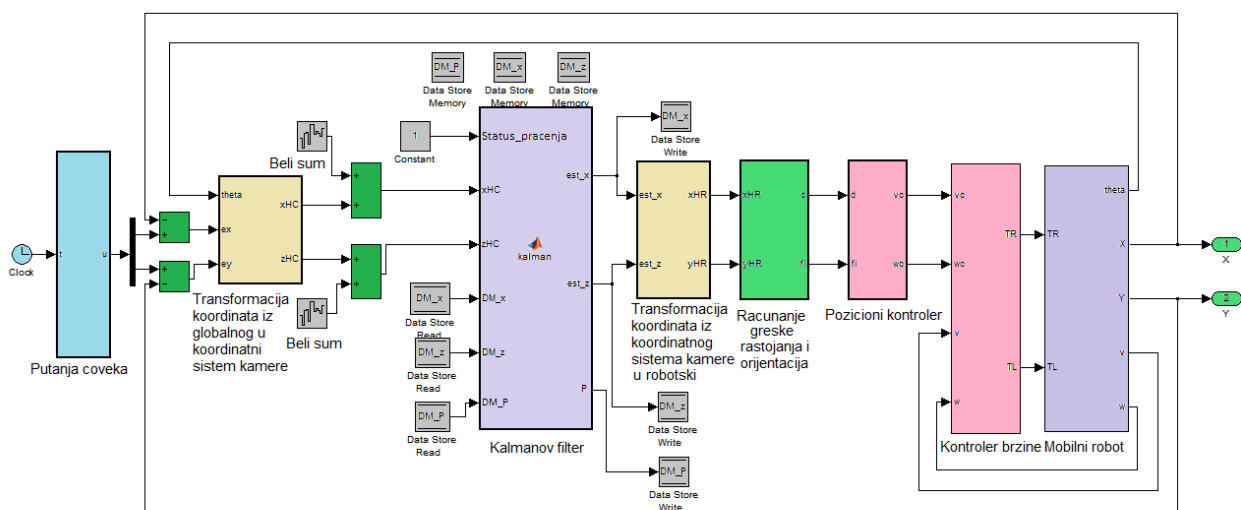
Pre izvođenja eksperimenta gde robot prati osobu u realnom vremenu, sistem je testiran simuliranjem robotskog sistema sa modulom praćenja gde robot prati čoveka i rezultati praćenja su prikazani u narednom poglavlju.

6.4. SIMULACIJA SISTEMA SA MODULOM PRAĆENJA ZASNOVANIM NA KALMANOVOM FILTERU

Simulaciona šema sistema sa modulom za praćenje je prikazana na slici 6.26. Modul praćenja je zvanovan na Kalmanovom filteru. Algoritam za modul praćenja je dat u **Dodatku B**. Na slici 6.27 je prikazana putanja čoveka i robota, pri čemu smo radi lakšeg testiranja sistema sa modulom praćenja, za cilj upravljanja uzeli da je rastojanje između robota i čoveka jednako nuli.

Na ovaj način se testira robusnost sistema vizije u odnosu na različite spoljašnje uticaje kao što su promena osvetljenja, promena ugla snimanja i prisustvo mnogo različitih objekata u sceni, kao i ponašanje mobilnog robota, tj. njegovog upravljačkog sistema.

Na slikama 6.28 i 6.29 prikazani su zašumljeni ulazi u modul praćenja, kao i izlazi iz tog modula. Na slici 6.29 uvećen je deo dijagrama radi jasnijeg prikaza rezultata estimacije. U cilju testiranja kretanje robota kada su ulazi u sistem zašumljeni, modul za prepoznavanja ponašanja čoveka je zanemaren, i izlaz iz modula praćenja se vodi direktno u upravljački modul.



Slika 6.26. Simulaciona šema sistema sa modulom za praćenje

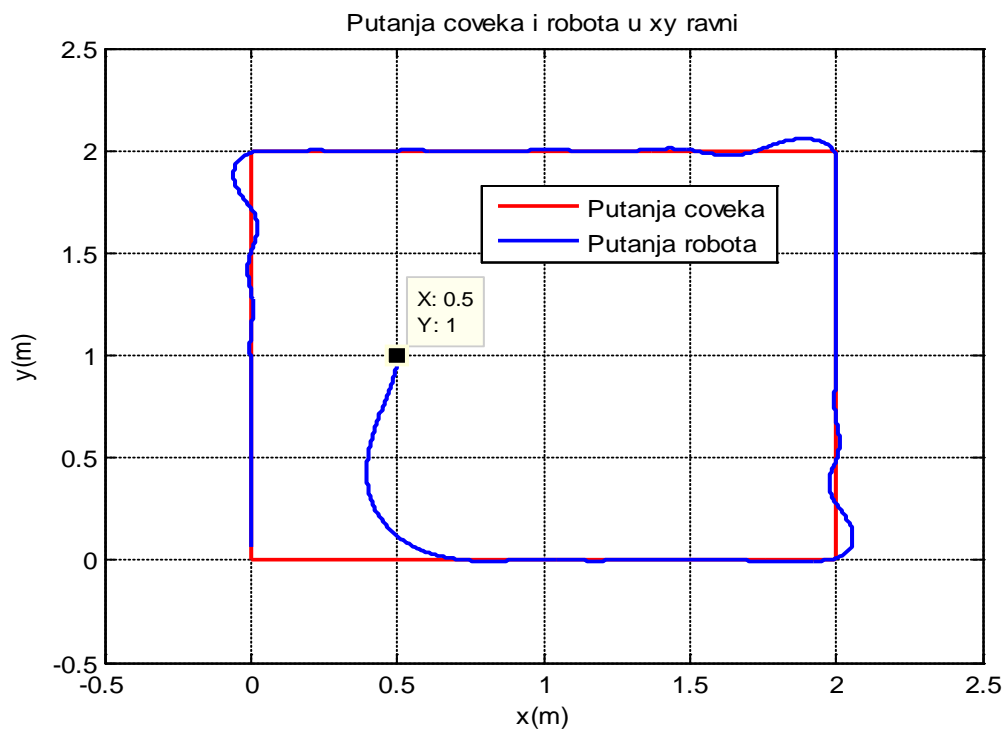
Transformacija koordinata položaja čoveka iz globalnog u koordinatni sistem kamere je urađena na isti način kao i transformacija koordinata iz globalnog u robotski koordinatni sistem što je opisano u poglavlju 4.3.2.

$$\begin{bmatrix} Z_{HC} \\ X_{HC} \end{bmatrix} = \begin{bmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{bmatrix} \begin{bmatrix} e_x \\ e_y \end{bmatrix} \quad (6.33)$$

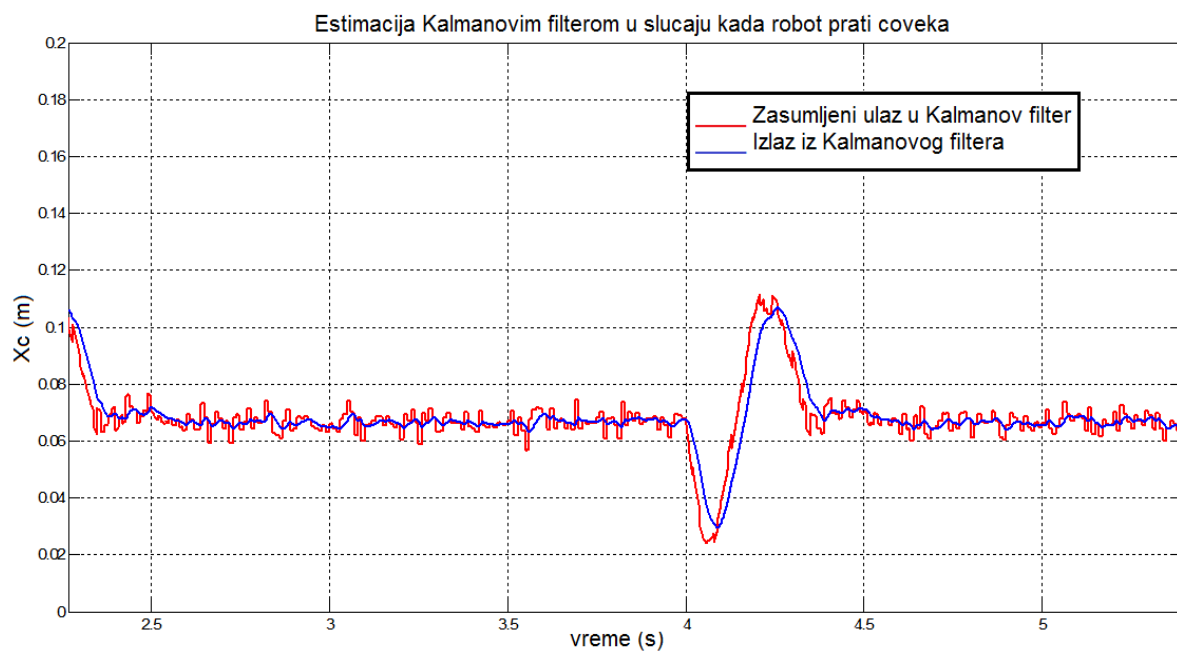
$$\begin{bmatrix} e_x \\ e_y \end{bmatrix} = \begin{bmatrix} x_{Human} - x_{robot} \\ y_{Human} - y_{robot} \end{bmatrix}$$

gde su x_{Human} i y_{Human} koordinate pozicije čoveka u globalnom kordinatnom sistemu, a x_{robot} i y_{robot} koordinate pozicije robota u globalnom kordinatnom sistemu.

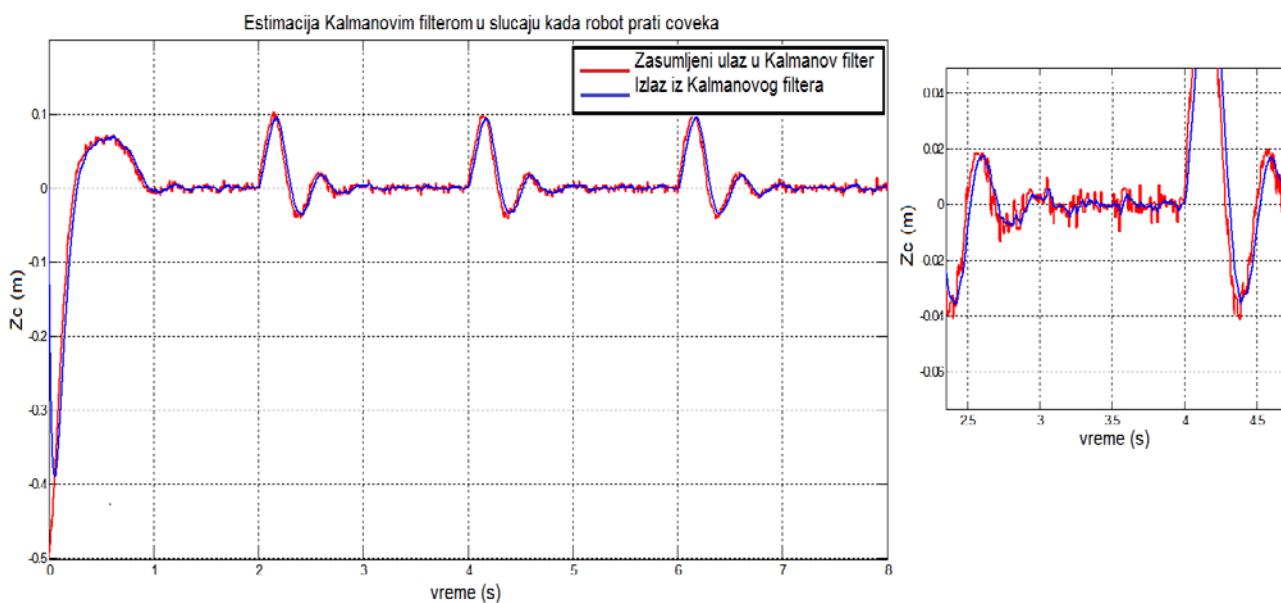
Memorijski blokovi (*eng.* Data memory block) su korišćeni da sačuvaju podatke o prethodnim stanjima dobijenim Kalmanovim filtrom, kako bi on-line Kalmanov filter u trenutku k+1 ocenio stanje na osnovu prethodnog stanja i podataka dobijenih vizijom.



Slika 6.27. Putanja čoveka i robota (robotski sistem sa modulom za praćenje)



Slika 6.28. Zašumljeni podaci o položaju čoveka u pravcu X_C ose i izlaz iz modula praćenja.



Slika 6.29. Zašumljeni podaci o položaju čoveka u pravcu Z_C ose i izlaz iz modula praćenja.

Na osnovu dobijenih rezultata može se zaključiti da modul praćenja zasnovan na Kalmanovom filteru, uspešno estimira položaj čoveka u horizontalnoj ravni, efikasno eliminiše poremećaje u sistemu i obezbeđuje pouzdane ulaze u upravljački modul.

7. UPRAVLJANJE MOBILNIM ROBOTOM NA OSNOVU INFORMACIJA SA VISOKOG NIVOVA UPRAVLJANJA

Prepoznavanje ponašanja čoveka se često koristi u čovek-robot interaktivnim aplikacijama, kao što su vuzuelno nadgledanje, rehabilitacija i robotika [122]. Prepoznavanje ponašanja čoveka na stvarnim video snimcima privlači sve veću pažnju istraživača koji se bave kompjuterskom vizijom. Veliki obim podataka, modernije metode za ekstraktovanje karakteristika (*eng. feature extraction*) i tehnike mašinskog učenja jesu među glavnim inovacijama u ovoj oblasti [123]. U literaturi postoji veliki broj metoda klasifikacije za prepoznavanje ponašanja čoveka kao što su metod potpornih vektora (*eng. support vector machines (SVM)*) [124], stablo odlučivanja (*eng. decision trees classifiers*) [125], klasifikator k-najbližih suseda (*eng. k-nearest neighbor (kNN) classifier*) [126], skriveni Markovljevi modeli (*eng. hidden Markov models*) [127], neuronske mreže [128] [129] kao i tzv. “data mining“ algoritmi kao što je naivni Bajesov klasifikator (*eng. Naïve Bayes classifier*) [130] i Bajesov mrežni klasifikator (*eng. Bayesian Network Classifiers*) [131].

Klasifikacija se generalno može podeliti na klasifikaciju koja se vrši nadgledano i koja se vrši nenadgledano. Kod nadgledane klasifikacije obučavanje klasifikatora se vrši na osnovu seta vektora kojima su unapred dodeljene oznake klase, pa se zatim klasifikator koristi za klasifikaciju novih neoznačenih vektora karakteristika. Kod nenadgledanog učenja klasifikator sam formira klase na osnovu sličnosti vektora karakteristika [132]. Postoji veoma veliki broj primena nadgledane klasifikacije u realnom životu, kao što je prepoznavanje lica, prepoznavanje potpisa, pronalaženje potrošača, detekcija spamova, biološki sistemi itd.

U zavisnosti od načina unosa ulaznih podataka koji su dati sistemu možemo klasifikovati različite klase. Eksperti svrstavaju skupove podataka (training setove) u određenom broju grupa (klasa ili oznaka). Svrstani podaci se zatim koriste za učenje modela (klasifikatora) u cilju klasifikacije novih, nepoznatih, podataka u prethodno definisane grupe.

Prepoznavanje ponašanja čoveka zasnovano na putanji kretanja takođe privlači mnogo pažnje. Drugim rečima, kvantitativne podatke o ljudskom kretanju potrebno je transformisati u kvalitativne, tako da se numeričke karakteristike kao što su brzina, pravac, itd., posmatrane osobe predstavljaju kao ponašanje, “kreće se prema nečemu” ili “kreće se od nečega” [133]. Kvalitativna prostorna reprezentacija i razumevanje su aktuelna oblast proučavanja koja se bavi, između ostalog, i formulacijom prostornog odnosa između dva fizička entiteta [134].

Predmet istraživanja u ovoj tezi, odnosi se na problem prepoznavanja ponašanja čoveka na osnovu informacija dobijenih od sistema robotske vizije koji kao senzor koristi kameru, u cilju izvršavanja zadataka u kojima se robot ponaša kao čovekov saradnik.

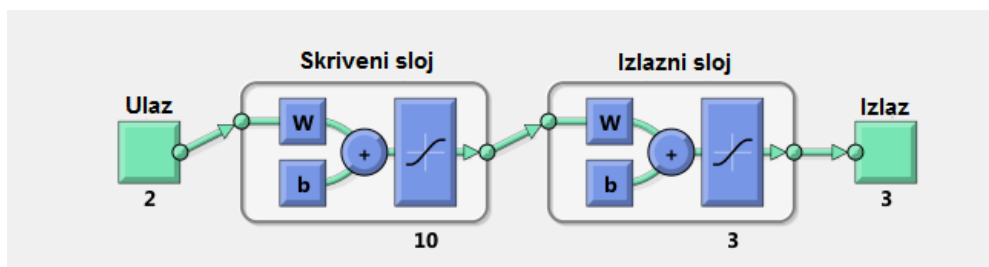
Da bi robot mogao da razume pokrete čoveka i da se ponaša u skladu s njima, neophodno je semantički prevesti podatke koje prikuplja senzor robotske vizije. Drugim rečima, kvantitativna informacija o poziciji čoveka se mora prevesti u kvalitativne podatke o ponašanju čoveka. U jednostavnom predstavljenom scenariju u kome robot prati čoveka saradnika na primer u istraživanju i inspekciji kontaminiranih/opasnih okruženja ili roboti-pratioci starijih ljudi, ponašanje čoveka treba biti predstavljeno kao “udaljava se od robota”, “stoji u odnosu na robot” i “kreće se prema robotu”. Za potrebe ove teze za klasifikaciju ponašanja čoveka razvijen je klasifikator zasnovan na neuronskim mrežama.

7.1. MODEL NEURONSKE MREŽE ZA KLASIFIKACIJU U CILJU PREPOZNAVANJA PONAŠANJA ČOVEKA

Neuronske mreže su se razvile kao važan alat za klasifikaciju. Prednost neuronskih mreža se ogleda u sledećim teoretskim aspektima:

- Kao prvo, neuronske mreže mogu da se prilagode ulaznim podacima bez bilo kakve naročite specifikacije funkcionalne ili distribucione forme za dati model.
- Drugo, neuronske mreže su univerzalni funkcionalni aproksimatori, što se ogleda u mogućnosti da neuronske mreže mogu da aproksimiraju bilo koju funkciju sa proizvoljnom tačnošću. Budući da bilo koja procedura klasifikacije traži funkcionalnu vezu između članova grupe i opisa objekta, tačna indentifikacija date funkcije je nesumnjivo važna.
- Treće, neuronske mreže su nelinearni modeli, što ih čini fleksibilnim u modelovanju kompleksnih veza u realnom svetu. Na kraju, neuronske mreže mogu da procene posteriorne verovatnoće, što pruža osnovu za postavljanje klasifikacionih pravila i obavljanje statističkih analiza [134].

U cilju klasifikacije ponašanja čoveka predložen je klasifikator zasnovan na neuronskoj mreži za prepoznavanje obrazaca sa 10 neurona u skrivenom sloju, čija je arhitektura prikazana na slici 7.1. U pitanju je neuronska mreža sa prostiranjem signala unapred sa jednim skrivenim slojem pri čemu su neuroni u skrivenom sloju sa sigmoidnom aktivacionom funkcijom.



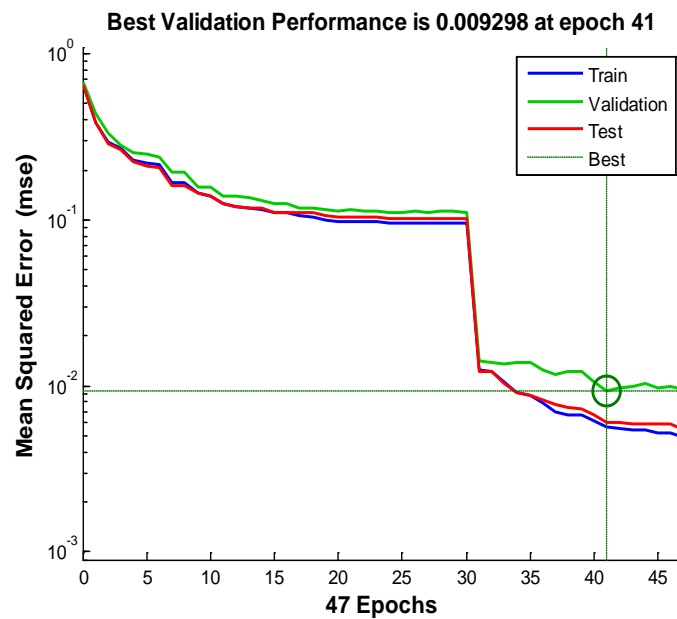
Slika 7.1. Arhitektura NN klasifikatora za prepoznavanje ponašanja ljudi

Kao ulaz klasifikatora predložene su greška rastojanje čoveka od robota d (robot treba da prati čoveka na rastojanju od 1.5m) i ugao orijentacije φ . Izlaz iz klasifikatora je verovatnoća da kretanje (ponašanje) čoveka koje ispitujemo pripada jednoj od tri klasa koje su predstavljene kao “udaljava se od robota”, “stoji u odnosu na robot” i “kreće se prema robotu”.

Za obučavanje, validaciju i testiranje neuronske mreže korišćeno je 1730 uzorka kojima su ručno određene labele. Slučajno je izabrano 70% setova podataka za obučavanje, a po 15% slučajno izabranih setova je korišćeno za validaciju i testiranje. Za obučavanje je korišćen skalirani konjugovani gradijent sa povratnom propagacijom greške (*eng.* scaled conjugate gradient backpropagation) koji ažurira težinske faktore i pripadnost stanja prema Levenberg–Marquardt optimizaciji, dok je srednje kvadratna greška MSE korišćena kao mera performanse tokom obučavanja mreže. Rezultati obučavanja, validacije i testiranja mreže mogu se videti na konfuzionim matricama prikazanim na slici 7.2. Na slici 7.3 je prikazana minimizacija greške kroz iteracije obučavanja neuronske mreže.



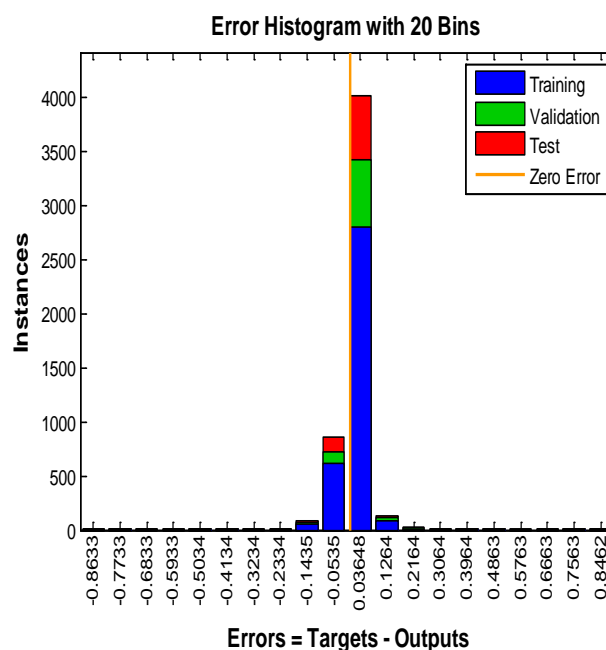
Slika 7.2. Konfuzionne matrice klasifikatora



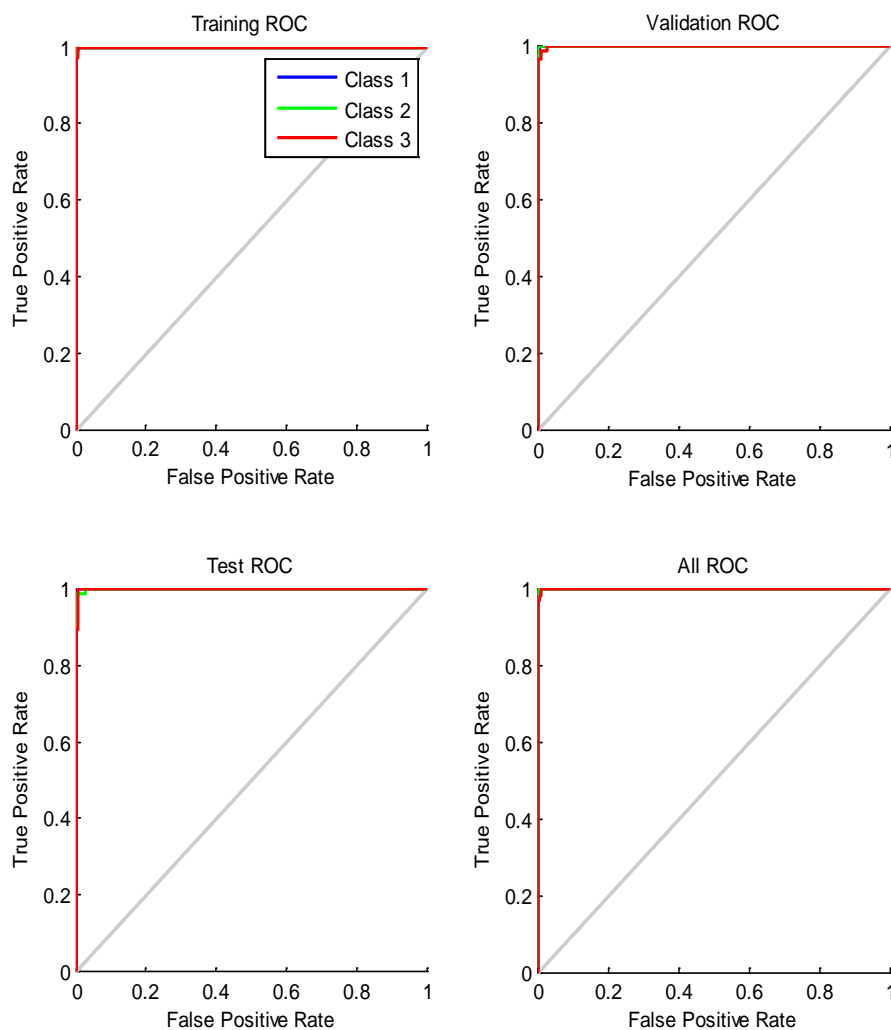
Slika 7.3. Minimizacija greške kroz iteracije obučavanja neuronske mreže

Obučavanje predloženog klasifikatora izvršeno je kroz 47 iteracija, a rezultati klasifikacije su zadovoljavajući. Tačnost obučenog klasifikatora je preko 99%, pa se može reći da predloženi klasifikator zasnovan na neuronskoj mreži daje dobre rezultate, i daje implementacijom ovakvog klasifikatora u viši nivo upravljanja moguće prepoznati ponašanje čoveka i u skladu s tim adekvatno upravljati kretanjem mobilnim robotom.

Na slikama 7.4 i 7.5 su prikazan histogram greške klasifikatora, odnosno radne karakteristike klasifikatora.



Slika 7.4. Histogram greške klasifikatora



Slika 7.5. Radne karakteristike klasifikatora

7.2. MODUL ODLUČIVANJA

Nakon klasifikaciju ljudskog ponašanja na visokom nivou modul odluke donosi odluke o načinu upravljanju mobilnim robotom na sledeći način:

- režim *praćenja*

U ovom slučaju, klasifikator daje podatke da se “čovjek udaljava od robota“ tj. da je greška rastojanja d pozitivna i rastei da se greška orijentacije φ povećava bilo u plusu ili minusu ili je približno jednaka nuli. Takođe, režim praćenja je aktivan i kada je greška rastojanja d približno jednaka nuli, ali greška orijentacije φ raste bilo u pozitivnom ili negativnom smeru, što znači da se čovek kreće oko robota.

U ovom slučaju pozicioni kontroler treba da sračuna linearnu i ugaonu brzinu prema formuli (4.21).

$$v_c = K_p d \cos \varphi$$
$$\omega_c = K_p \sin \varphi \cos \varphi + K_a \varphi$$

- režim *stajanja*

U slučaju kada je izlaz iz klasifikatora “stoji u odnosu na robot”, greška rastojanja d i greška orijentacije φ približno jednake nuli.

U ovom slučaju vrednost linearne i ugaone brzina koju šalje pozicioni kontroler je jednaka nuli.

$$v_c = 0$$
$$\omega_c = 0$$

- režima *prilaženja*

Kod režima prilaženja izlaz iz klasifikatora je “kreće se prema robotu“, što znači da greška rasrojanja d raste u negativnom smeru dok greška orijentacije nije bitna (greška orijentacije φ može da bude pozitivna, negativna ili približno jednaka nuli). U radnom scenariju znači da čovek želi nešto da stavi na robotsku platformu, pa prema tome robot treba da stane i “pusti” čoveka da mu priđe. Kada se čovek približava robotu vidno polje kamere se smanjuje i zbog toga je neophodno da sve vreme bude aktivno upravljanje orijentacijom robota, da kamera ne bi izgubila čoveka, tj. da se čovek uvek nalazi duž Z_c ose kamere.

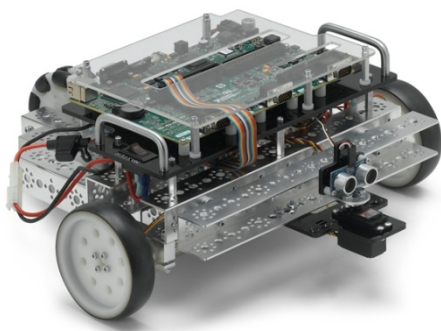
Vrednost linearne i ugaone brzina koju šalje pozicioni kontroler treba da bude sledeća.

$$v_c = 0$$
$$\omega_c = K_p \sin \varphi \cos \varphi + K_a \varphi$$

7.3. INTEGRACIJA DANI MOBILNOG ROBOTA SA SISTEMOM ROBOTSKE VIZIJE

Za potrebe ove teze, na Mašinskom fakultetu, upravljanje višeg i srednjeg nivoa, testirano je na robotskoj platformi NI Robotics Starter Kit 2.0, gde ASUS Xtion 3D senzor pruža vizuelne informacije o okolini robota (Slika 6.7).

Pokretna robotska platforma NI Robotics Starter Kit 2.0, poznat kao DaNI robot (Slika 7.1), je razvijen od strane kompanije National Instruments. Robot poseduje senzore, motore i NI Single-Board RIO-9632 kontrolnu ploču sa procesorom za rad u realnom vremenu. Korišćeni mobilni robot poseduje dva točka koji se pogone sa dva motora i jedan pomoćni točak bez pogona. Upravljanje na najnižem nivou je već rešeno od strane National Instruments-a i kinematikom (pozicijom) robota se može upravljati korišćenjem LabVIEW Robotics modula.



Slika 7.1: DaNI robot – Starter Kit 2.0 [114]

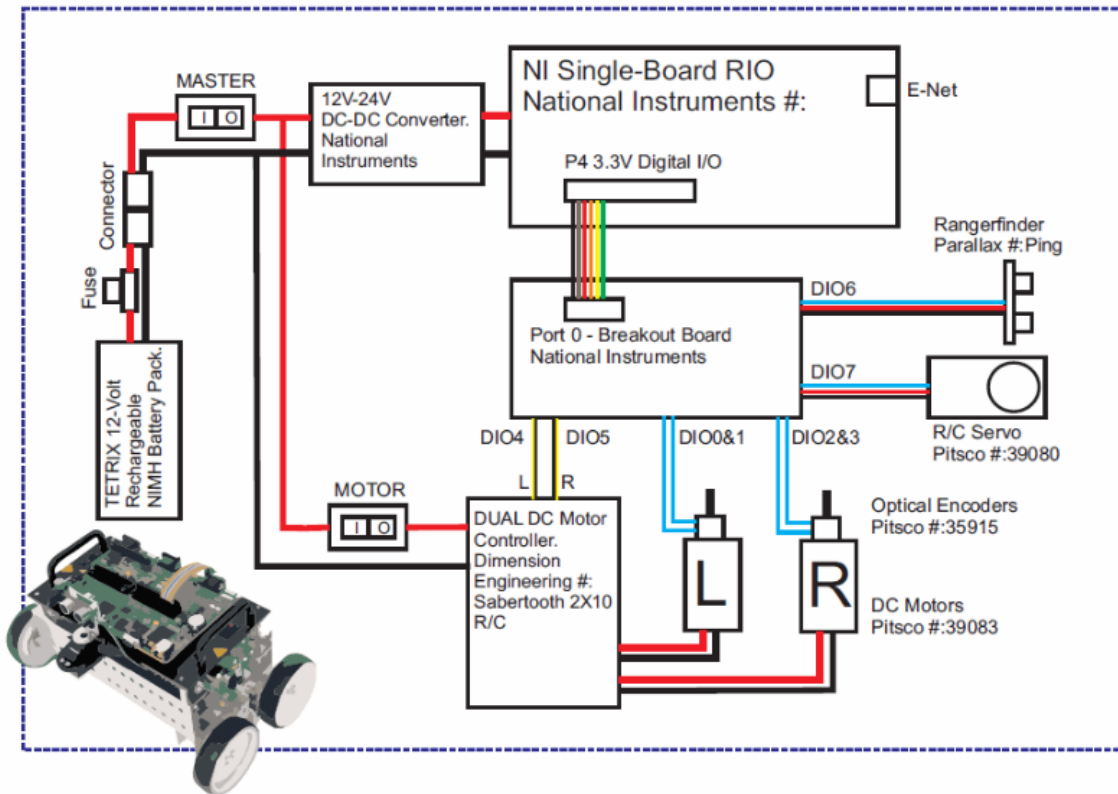
Specifikacija DaNI robota i senzora i aktuatora koje poseduje je data u tabeli 7.1

Tabela 7.1: Specifikacija DaNI robota [113]

| | |
|---------------------------------|--|
| Veličina platforme | 405 mm x 368 mm x 150 mm |
| Masa | 3.6 kg |
| Pitsco Education DC Motori | Napajanje 12V, Brzina 152 rpm, Moment 2.12 N.m. |
| Optički enkoderi | Napajanje 5V, Broj impulsa po obrtaju 400 |
| Ultrazvučni senzor | Napajanje 5V, Domet od 2cm do 3m |
| Pitsco Education TETRIX točkovi | 2 točka + 1 slobodan točak, 101 mm prečnik točka |

Na slici 7.2 prikazan je blok dijagram LabVIEW Starter Kit-a.

LabVIEW Robotics Starter Kit (Block Diagram)



Slika 7.2: Blok dijagram LabVIEW Starter Kit-a [113]

Asus Xtion PRO LIVE 3D senzor je povezan sa industrijskim računarom (Slika 7.3) koji je postavljen na DaNI mobilnu robotsku platformu, a komunikacija industrijskog računara i robotske platforme je ostvarena preko Lan kabela (Slika 6.7). Matworks obezbeđuje drajvere za konekciju Asus Xtion PRO LIVE 3D senzora sa Simulinkom koristeći OpenNi softver.

Specifikacije industrijskog računara korišćenog za potrebe ove teze su date u tabeli 7.2. Asus Xtion PRO LIVE 3D sensor je detaljno opisan u poglavlju 6.2.

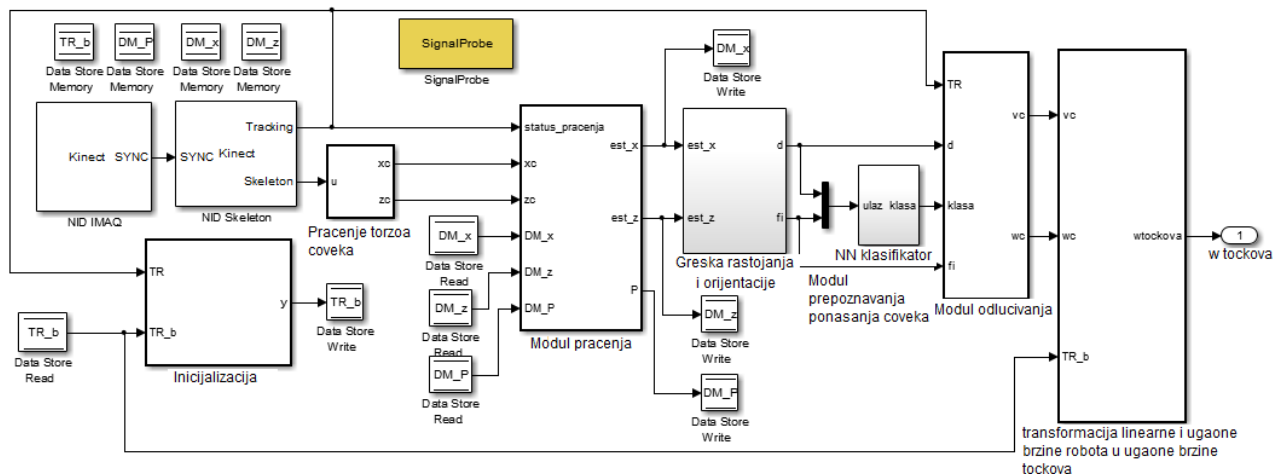


Slika 7.3: Industrijski PC [115]

Tabela 7.2: Specifikacije industrijskog računara [115]

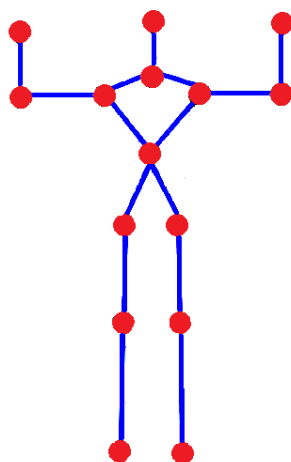
| | | | |
|--------------------|---|-------------------|--|
| Model | AES-HM76Z1FLi71 | SIM | 1 x SIM slot (SIM card zamenjiva bez otvaranje kućišta) |
| Procesor | Intel Core i7 3517UE 1.7GHz | Ekspanzija | 1 x Mini PCIe socket(For 3.5G/WiFi) |
| Čipset | Intel HM76 | USB | 3 x USB 3.0 2 x USB 2.0 (1 za Mini PCIe / 1 za Combo konektor) |
| Memorija | 2 x DDR3L-1600MHz, SO-DIMM, up to 16GB | Serijski port | 2 x RS-232 (DB9 konektor) 2 x RS-422/485 (DB15 konektor, podeljen sa GPIO) |
| Bios | AMI | Audio | 1 x Line-Out (Combo konektor) Realtek ALC662 |
| Grafički kontroler | Integrisan sa HM76 | GPIO | 8 x GPI (DB15 konektor, podeljen sa RS-422/485) 8 x GPO (DB15 konektor, podeljen sa RS-422/485) |
| Video | 2 x HDMI (HDMI Type A connector) 1 x VGA (Combo connector) | Napajanje | 12VDC Single Voltage Input |
| Ethernet | 2 x GbE (RJ45 sa LED) Realtek RTL8111F | Operativni system | Windows 7 (32/64) Fedora 17 Ubuntu |

Simulink model za prikupljanje podataka i upravljanje pozicijom DANI robotskom mobilnom platformom, sa modulima za praćenje i prepoznavanje ponašanja čoveka i modulom odlučivanja, je prikazan na slici 7.4.



Slika 7.4: Simulink model visokog i srednjeg nivoa upravljanja DANI robotskom mobilnom platformom i akviziciju podataka Asus Xtion PRO LIVE 3D senzorom

Budući da OpenNi zahteva inicijalizaciju podizanjem ruku kao što je prikazano na slici 7.5 u Simulink modelu za akviziciju i upravljanje je dodat još jedan blok koji služi da se robot ne kreće sve dok prvi put ne dobije validne podatke sa 3D senzora robotske vizije. Memorijski blok TR_b služi kao brojač, tj. njegova inicijalna vrednost je 0 i ostaje 0 sve dok je status praćenja jednak nuli, tj. sve dok kamera prvi put ne detektuje čoveka. Kada status praćenja postane jednak jedinici, TR_b dobija vrednost veću od 0 i zadržava je do kraja praćenja. U slučaju da vizija usled poremećaja u sistemu ne šalje podatke o poziciji čoveka, tj. status praćenja je ponovo jednak nuli, modul praćenja će “predvideti” poziciju čoveka i na taj način će obezbediti da sistem uvek ima adekvatne ulaze u upravljački modul.

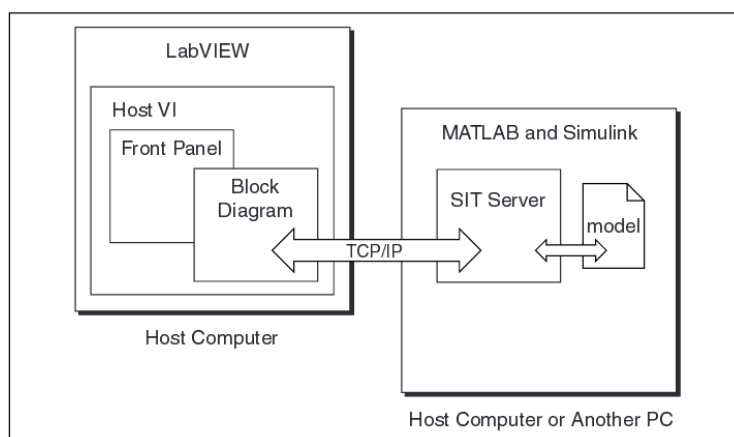


Slika 7.5: Inicijalizacija praćenja Asus Xtion PRO LIVE 3D senzorom

Da bi se upravljalo DANI robotom potrebno je prvo uspostaviti vezu između Simulink-a i LabView-a. Za konekciju između Matlaba i LabView-a, National Instruments je obezbedio set alata (*eng.* toolkit) pod nazivom LabVIEW Simulation Interface Toolkit. The Simulation Interface Toolkit (SIT) automatski generiše LabVIEW kod za povezivanje sa Simulink modelom što omogućava fleksibilno i jednostavno korišćenje korisničkog interfejsa. LabVIEW razmenjuje podatke sa Matlab-om koristeći TCP/IP protokol. Nakon instaliranja SIT-a, u Matlabu se pokreće NISIT_AddPath.m skripta, koja generiše SIT Server biblioteku, a zatim se pokreće konekcija sa LabVIEW-om, ukucavanjem komande NISITServer('start',6011). SignalProbe blok iz SIT Server biblioteke se dodaje Simulink modelu da bi se omogućilo slanje podataka u LabVIEW. Mana SIT-a je to što podržava samo 32 bitnu verziju Matlaba.

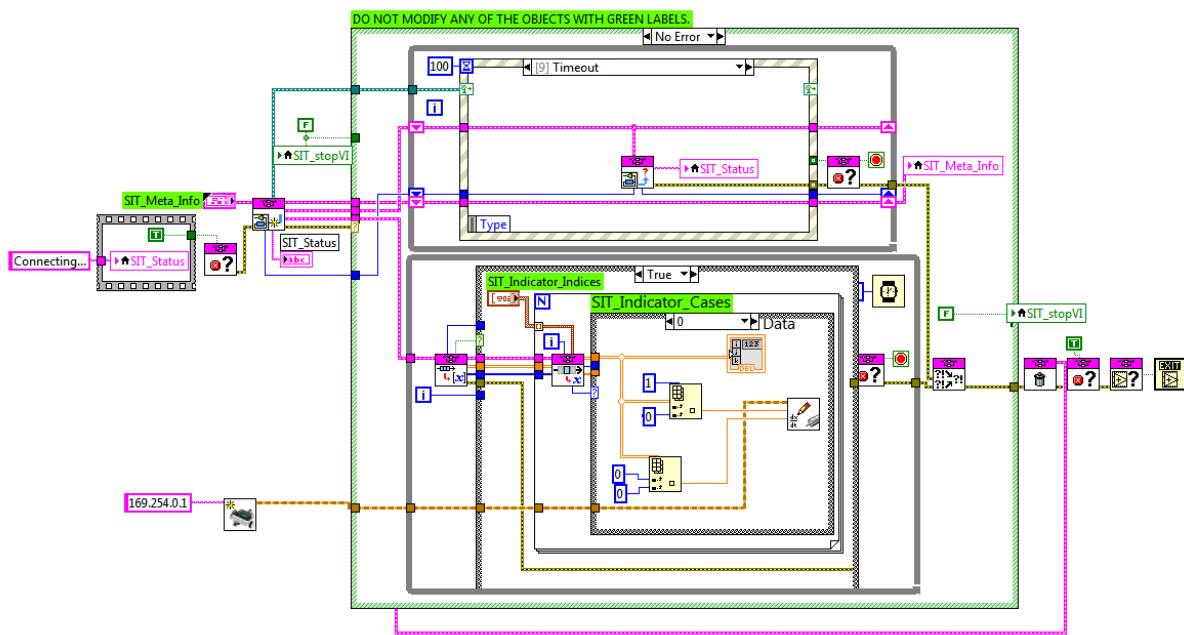
Da bi se pokrenuli korekcija sa NISIT Server u LabVIEW-u, koristi se SIT Connection Manager u kome se bira ime Simulink modela kojeg želimo da pokrenemo.

Na slici 7.6 je prikazan šematski dijagram veze.



Slika 7.6: Pokretanje Simulink modela u LabView-u [136]

Na osnovu ove veze može se upravljati DaNI robotskom platformom u cilju praćenja čoveka u random scenariju. LabView algoritam za upravljanje DaNI robotskom platformom na osnovu podataka dobijenih sa viših hijerarhijskih nivoa upravljanja je prikazan na slici 7.7.

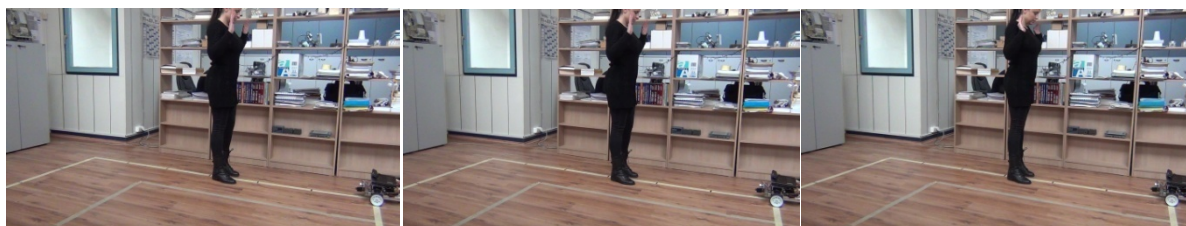


Slika 7.7: LabView algoritam za upravljanje DaNI robotskom platformom

7.4. PRAĆENJE ČOVEKA MOBILNOM ROBOTSKOM PLATFORMOM DANI SA ASUS XTION 3D SENZOROM

Testiranje praćenja čoveka mobilnom robotskom platformom DANI sa ASUS XTION 3D senzorom je rađeno u radnom scenariju gde DANI prati čoveka koji se kreće po pravougaonoj duži koja je nacrtana na podu i podeljena je markerima, koji su na međusobnom rastojanju od 0.5 m, radi lakšeg vizualnog praćenja rastojanja između robota i čoveka. Stacionarnom kamerom snimano je kretanje robota i čoveka.

U prvom eksperimentu testiran je režim praćenja. Čovek se udaljavao od robota, dok ga je robot pratio održavajući konstantno rastojanje od 1.5 m. Na slici 7.8 su prikazni rezultati praćenja u vidu niza slika uzetih na svakih 30 frejmova sa video snimka snimljenog stacionarnom kamerom. Pre početka praćenja, izvršena je inicijalizacija sistema vizije podizanjem ruku.

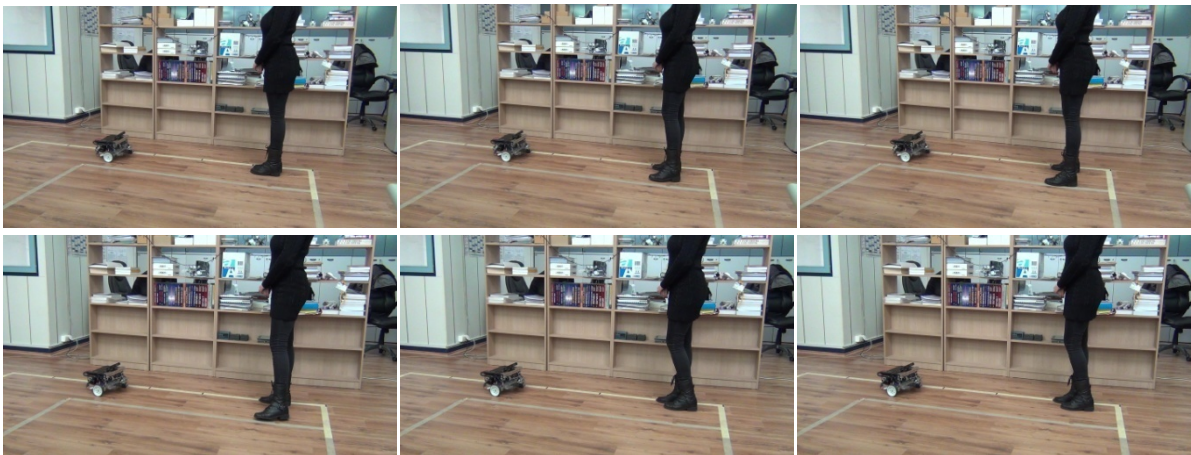






Slika 7.8: Rezultati eksperimenta režima praćenja

U drugom eksperimentu testiran je režim stajanja. Čovek se stajao u odnosu na robot, nakon čega se robot pozicionirao na rastojanju od 1.5 m. Na slici 7.9 su prikazni rezultati ovog eksperimenta u vidu niza slika uzetih na svakih 30 frejmova sa video snimka snimljenog stacionarnom kamerom.



Slika 7.9: Rezultati eksperimenta režima stajanja

U trećem eksperimentu testiran je režim prilaženja. U ovom eksperimentu, čovek se približavao robotu, pri čemu je robot “dozvolio” čoveku da mu priđe. Tek kada se je čovek udaljio na više od 1.5 m robot je počeo da prati čoveka. Na slici 7.10 su prikazni rezultati ovog eksperimenta u vidu niza slika uzetih na svakih 30 frejmova sa video snimka snimljenog stacionarnom kamerom.





Slika 7.10: Rezultati eksperimenta režima prilaženja

8. ZAKLJUČCI I DALJA ISTRAŽIVANJA

Istraživanja sprovedena za potrebe ove doktorske disertacije obuhvataju više aktuelnih naučno-istraživačkih pravaca koji se odnose na praćenje ljudi pomoću mobilne robotske platforme sa senzorom robotske vizije.

U ovoj tezi su prezentovani i analizirani različiti algoritmi, kao što su Kalmanovi i Partikle filteri, za rešavanje problema praćenja ljudi na osnovu podataka dobijenih sistemom robotske vizije. Značajan doprinos u rešavanju problema praćenja ljudi na osnovu informacija dobijenih senzorom robotske vizije, postignut je integracijom filtera u modul vizije. Integracija filtera u modul vizije se koristi u cilju praćenja ljudi kada vizija ne daje podatke o poziciji čoveka, usled poremećaja u sistemu, kao što su promena osvetljenja, promena ugla snimanja i prisustvo mnogo različitih objekata u sceni, čime je postignut robusan sistem robotske vizije. Ovakav sistem se sastoji od dva sub-modula, modula vizije za detekciju čoveka i modula praćenja, koji obezbeđuje da robot dobija ulazne podatke kontinualno kako bi adekvatno upravljao svojim kretanjem. Ono što je značajno je da se ova struktura za estimaciju položaja čoveka može koristiti sa različitim mobilnim platformama. Takođe, izvršena je uporedna analiza algoritama za rešavanje problema detekcije i praćenja ljudi, u cilju pronalaženja najefikasnijeg estimatora za estimiranje/predikciju pozicije čoveka.

Razvijeni modul za praćenje položaja čoveka testiran je na Institutu za Automatiku (IAT) Univerziteta u Bremenu primenom sistema stereo vizije za detekciju čoveka koji koristi Point Grey Bumblebee XB3 stereo kameru, kao i na Mašinskom fakultetu Unverzitata u Nišu primenom sistema vizije koji koristi OpenNI biblioteku i ASUS Xtion 3D senzor.

Primenom savremenih metoda iz domena mašinskog učenja kao što su neuronske mreže, razvijen je modul za prepoznavanje ponašanja čoveka koji obezbeđuje inteligentnu interakciju mobilne robotske platforme sa okolinom.

U okviru ove teze razvijeno je i generičko simulaciono okruženje koje predstavlja podršku za razvoj i primenu stvarnog (*eng.* real-world) upravljačkog sistema. Ovakvo simulaciono okruženje može lako biti korišćeno od strane drugih istraživača, kao i za druge mobilne robote uz odgovarajuće modifikacije.

Razvijena je hijerarhijska struktura upravljanja koja na visokom novou prati i prepoznaje ponašanje čoveka i generiše ulaze za niže nivoe upravljanja u cilju izvršavanja zadataka u kojima se robot ponaša kao čovekov saradnik.

U hijerarhijskoj strukturi upravljanja razvijenoj u ovoj tezi, moduli za detekciju, praćenje i predikciju i prepoznavanje ponašanja čoveka, kao i modul odlučivanja su na visokom nivou

strukture upravljanja (*eng.* high-level control). Upravljanje pozicijom i orijentacijom mobilnog robota se svrstava u srednji nivo upravljanja (*eng.* medium level). Upravljački sistem niskog nivoa (*eng.* low level) sastoji se od multivarijabilnog PD kontrolera koji je odgovoran za konvertovanje brzina na izlazu pozicionog kontrolera u obrtni moment točkova robota, da bi se obezbedilo praćenje ulazne referentne brzine od strane robota. Razvijeni upravljački algoritmi se mogu primeniti i u drugim oblastima robotike, kao što je praćenje unapred zadate putanje mobilnim robotom, itd.

Eksperimentalna evaluacija predloženog hijerarhijskog upravljanja urađena je na Mašinskom fakultetu, Univerziteta u Nišu, korišćenjem 3D senzora ASUS Xtion PRO LIVE i DaNI mobilnog robota firme National Instruments, koji u scenariju laboratorije prepoznaje i prati čoveka i prepoznaje njegovo ponašanje.

Prezentirani naučni doprinosi, koji pored nesumnjivog praktičnog značaja, predstavljaju i proširenje postojećih saznanja u rešavanju kompleksnih problema prepoznavanja i praćenja ljudi, kao i njihovog ponašanja u robotskoj viziji, koje je često veoma nepredvidivo, kao i sveobuhvatno uvođenje i integraciju robota u scenarija gde pomažu čoveku u obavljanju različitih poslova u interakciji sa robotom. Pri tom, teorijski doprinosi nisu od značaja samo za prepoznavanje i praćenje ljudi robotskom vizijom, već i za prepoznavanje i praćenje različitih objekata u pokretu u različitim aplikacijama kompjuterske vizije, kao što su napredni sistemi za asistenciju vozačima u cilju povećanja sigurnosti u saobraćaju ili sigurnosni sistemi.

Pravci daljeg istraživanja biće usmereni ka oblasti planiranja putanje (*eng.* Path-planning), u cilju razvoja algoritama za detekciju i obilaženje prepreka koji se mogu naći na putanji kretanja mobilnog robota dok prati čoveka.

LITERATURA

- [1] Takafumi S, Takashi Y, Manabu N, Hideichi N, Seiji T, Nobuto M (2008), **Person Following Robot with Vision-based and Sensor Fusion Tracking Algorithm**, *Computer Vision, Book edited by: Xiong Zhihui*, pp. 538, I-Tech, Vienna, Austria
- [2] Borenstein J, Thomas D, Sights B, Ojeda L, Bankole P, Fellars D (2010) ,**Human Leader and Robot Follower Team: Correcting Leader's Position from Follower's Heading**, *Proc. of the SPIE Defense, Security+Sensing; Unmanned system technology XII; Conf. DS117, Unmanned, Robotics, and layered Systems*, Orlando, FL.
- [3] Satake J., Miura J.,(2009), **Robust Stereo-Based Person Detection and Tracking for a Person Following Robot**, *Proceedings of the IEEE ICRA 2009, Workshop on People Detection and Tracking*, Kobe, Japan.
- [4] Munder S., Schnorr C., Gavrila D.M.,(2008),**Pedestrian Detection and Tracking Using a Mixture of View-Based Shape-Texture Models**, *IEEE Trans. on Intelligent Transportation Systems*, Vol. 9, No. 2, pp. 333–343.
- [5] Han B., Joo S.W., Davis L.S., (2007), **Probabilistic Fusion Tracking Using Mixture Kernel-Based Bayesian Filtering**, *In Proceedings of the 11th Int. Conf. on Computer Vision*.
- [6] Viola P., Jones M.J., Snow D., (2005),**Detecting Pedestrians Using Patterns of Motion and Appearance**, *Int. J. of Computer Vision*, Vol. 63, No. 2, pp. 153–161, 2005.
- [7] Dalal N., Briggs B.,(2005),**Histograms of Oriented Gradients for Human Detection**, *In Proceedings of 2005 IEEE Conf. on Computer Vision and Pattern Recognition*, pp. 886–893.
- [8] Leu A., (2014), **Robust Real-time Vision-based Human Detection and Tracking**, *PhD thesis* , Universität Bremen.
- [9] Beymer D., Konolige K.,(1999), **Real-Time Tracking of Multiple People Using Continuous Detection**, *In Proceedings of the 7th Int. Conf. on Computer Vision*.
- [10] Microsoft Kinect product description [online] Available at: <<http://www.microsoft.com/enus/kinectforwindows/>> [Accessed January 2013]
- [11] Asus Xtion PRO LIVE product description [online] Available at: <http://www.asus.com/Multimedia/Xtion_PRO_LIVE/> [Accessed May 2013]
- [12] Open NI homepage [online] Available at: <<http://www.openni.org/>> [Accessed May 2013]

- [13] Burishka D., Geiman J., Hager G., (2003), **Optimal landmark configuration for vision-based control of mobile robot**, *Proceedings of the 2003 IEEE International Conference on Rob. and Auto.*, Taipei, Taiwan.
- [14] Cowan N.J., Koditschek D.E., (1999), **Planar image based visual servoing as a navigation problem**, *Proceedings of the 1999 IEEE International Conference on Rob. and Auto.*, Detroit, Michigan.
- [15] DeSouza G.N., Kak A.C.,(2002),**Vision for mobile robot navigation: a survey**, *IEEE Transaction on Pattern Analysis and Machine Intelligence*, vol. 24.
- [16] Volosyak I., Ivlev O., Graeser A., (2005),**Rehabilitation robot FRIEND II - the general concept and current implementation**,*In Proc. of the IEEE 9th Int. Conf.on Rehabilitation Robotics ICORR 2005*, pp. 540-544, Chicago, USA.
- [17] Grigorescu S., Ristić-Durrant D., Vuppala K. S., Axel G., (2008),**Closed-Loop Control in Image Processing for Improvement of Object Recognition**, *Proceedings of the 17th World Congress The International Federation of Automatic Control*, Seoul, Korea.
- [18] Hericks, M., Krebs, U., Kuzmicheva, O., (2011),**A Mobile Reconnaissance Robot for Investigation of Dangerous Sites**, *The 2011 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2011)*, San Francisco, California.
- [19] Petrović E., Leu A., Ristić-Durrant D., Nikolić V., (2013), **Stereo-Vision Based Human Tracking for Robotic Follower**, *International Journal of Advanced Robotic Systems*, DOI: 10.5772/56124.
- [20] Leu A., Aiteanu D., Gräser A., (2012),**High Speed Stereo Vision Based Automotive Collision Warning System**,*Applied Computational Intelligence in Engineering and Information Technology Volume 1 of the series Topics in Intelligent Engineering and Informatics*,pp 187-199.
- [21] Leu A., Aiteanu D., Gräser A., (2011),**A novel stereo camera based collision warning system for automotive applications**,*6th IEEE International Symposium on Applied Computational Intelligence and Informatics*, Timișoara, Romania
- [22] Hartley R., Zisserman A., (2003)**Multiple View Geometry in computer vision**, Cambridge University Press.
- [23] Ristić-Durrant, 3D Reconstruction, Lecture X, Univerzitat Bremen.
- [24] Natarajan S K, Ristić-Durrant D, Leu A, Graeser A, (2011), **Robust stereo-vision based 3D modeling of real-world objects for assistive robotic applications**,*In Proc of the IEEE Int. Conf. on Intelligent RObots and Systems*, pp 786-792.
- [25] Hu M. K (1962) ,**Visual Pattern Recognition by Moment Invariants**, *IRE Transaction on Information Theory*.

- [26] Zang G.P., **Neural Networks for Classification: A Survey**, (2000), IEEE Transactions on System, Man and Cybernetics- Part C: Application and Reviews, vol.30, No. 4.
- [27] K. Hornik, (1991), **Approximation capabilities of multilayer feedforward networks**, *Neural Networks*, vol. 4, pp. 251–257.
- [28] Marchewka D., Piątek M., **Wheeled mobile robot modeling aspects**, *Proceedings of the international conference on Robotics in Education* , 16–17.9. 2010 Bratislava, Slovakia , ISBN: 978-80-2773353-3, pp. 213–216.
- [29] Marinho T., Ramos J.L., (2003), **Model and Control of a differential drive mobile robot**, <http://web.mit.edu/jlramos/www/Arquivos/ReportDifferentialDrive.pdf>.
- [30] Dhaouadi R., Abu Hatab A., (2013), **Dynamic Modelling of Differential-Drive Mobile Robots using Lagrange and Newton-Euler Methodologies: A Unified Framework**, *Advances in Robotics and Automation*, Vol.2, No. 2, <http://dx.doi.org/10.4172/2168-9695.1000107>.
- [31] Yun X., Yamamoto Y., (1993), **Internal dynamics of a wheeled mobile robot**, *IEEE/RSJ International Conference on Intelligent Robots and System* ,pp. 1288-1294.
- [32] Fukao T., Nakagawa H., Adachi N, (2000), **Adaptive Tracking Control of a Nonholonomic Mobile Robot** , *IEEE Transaction on Robotics and Automation Vol. 16*, pp. 609-615.
- [33] Hou Z.G., Zou A.M., Cheng L., Tan M., (2009), **Adaptive Control of an Electrically Driven Nonholonomic Mobile Robot Via Back stepping and Fuzzy Approach**, *IEEE Transaction on Control Systems Technology*, Vol. 17, pp. 803-815.
- [34] Lewis F.R, (1997), **Control of a nonholonomic mobile robot: backstepping kinematics into dynamics**, *Journal of Robotic Systems*, Vol. 14, pp. 149-163.
- [35] Yamamoto Y., Yun X., (1992), **On Feedback Linearization of Mobile Robots**, *Technical Report No. MS- CIS-92-45, Philadelphia, PA*.
- [36] Sarkar N., Yun X., Kumar V., (1994) , **Control of mechanical systems with rolling constraints: Application to dynamic control of mobile robots** , *The International Journal of Robotics Research*, Vol. 13, No. 1, pp. 55-69.
- [37] De Santis R.M., (1995), **Modeling and Path-tracking Control of a mobile Wheeled Robot with a Differential Drive**, *Robotica*, Vol.13, pp. 401-410.
- [38] Albagul A., Wahyudi A., (2004), **Dynamic Modeling and Adaptive Traction Control for Mobile Robots**, *International Journal of Advanced Robotic Systems*, Vol. 1, pp.149-154.
- [39] Mohareri O., **Mobile robot trajectory tracking using neural network**, (2009), PhD Thesis, Presented to the faculty of the American University of Sharjah, College of Engineering .

- [40] Thanjavur K., Rajagopalan R., (1997), Ease of dynamic modeling of wheeled mobile robots (WMRs) using Kane's approach.
- [41] Bloch A. M., (2003), **Nonholonomic Mechanics and Control** (Book), *Interdisciplinary Applied Mathematics*, Vol. 24, ISBN: 978-1-4419-3043-9 (Print) 978-0-387-21644-7 (Online).
- [42] Aung W.P, (2007),**Analysis on Modeling and Simulink of DC Motor and its Driving System Used for Wheeled Mobile Robot**,*World Academy of Science, Engineering and Technology International Journal of Electrical, Computer, Energetic, Electronic and Communication Engineering* ,Vol.1, No.8, pp.1141-1148.
- [43] Khatib, O., 1986. **Real-time obstacle avoidance for manipulators and mobile robots**. *Int. J. Robot. Res.*, 5(1): 90-98.
- [44] Mabrouk, M.H. and C.R. McInnes, 2008. Solving the potential field local minimum problem using internal agent states. *Robot. Auton. Syst.*, 56(12): 1050-1060.
- [45] Karim Benbouabdallah and Zhu Qi-Dan, Improved Genetic Algorithm Lyapunov-Based Controller for Mobile Robot Tracking a Moving Target, *Research Journal of Applied Sciences, Engineering and Technology* 5(15): 4023-4028, 2013,ISSN: 2040-7459; e-ISSN: 2040-7467.
- [46] Li, T.H.S., S.J. Chang and W. Tong, 2004. Fuzzy target tracking control of autonomous mobile robots by using infrared sensors. *IEEE T. Fuzzy Syst.*, 12(4): 491-501.
- [47] Sašo Blaižić, Miguel Bernal, Trajectory Tracking for Nonholonomic Mobile Robots based on Extended Models, Preprints of the 18th IFAC World Congress Milano (Italy) August 28 - September 2, 2011.
- [48] Turki Y. Abdalla Mustafa I. Hamzah, Trajectory Tracking Control for Mobile Robot using Wavelet Network, *International Journal of Computer Applications* (0975 – 8887)Volume 74– No.3, July 2013.
- [49] Moreno, L., J.M. Armingol, S. Garrido, A.D.L. Escalera and M.A. Salichs, 2002. Genetic algorithm for mobile robot localization using ultrasonic sensors. *J. Intell. Robot. Syst.*, 34(2): 135-154.
- [50] Qiuling, J., X. Xing and G. Li, 2007. Formation path tracking controller of multiple robot system by high order sliding mode. *Proceeding of IEEE International Conference on Automation and Logistics*, pp: 923-92.
- [51] Guldner J, Utkin VI. Stabilization of nonholonomic mobile robot using Lyapunov functions for navigation and sliding mode control. *Control – Theory Adv Technol* 1994;10(4):635–47.]

- [52] Sandeep Kumar Malu ,Jharna Majumdar, Kinematics, Localization and Control of Differential Drive Mobile Robot, Global Journal of Researches in Engineering:Robotics & Nano-Tech Volume 14 Issue 1 Version 1.0 Year 2014.
- [53] Aicardi, M., G. Casalino, A. Bicchi and A. Balestrino, 1995. Closed loop steering of unicycle-like vehicles via Lyapunov techniques. *IEEE Robot. Autom. Mag.*, 2(1): 27-35.
- [54] Rudzuan Mohd Nor, Hazry Desa, Mohd Sofian M. Rosbi and R.Nagarajan, Developing an Autonomous Mobile Robot and a Study of Navigation Towards Nonholonomic Problems. Proceedings of the International Conference on Man-Machine Systems (ICoMMS) 11 – 13 October 2009, Batu Ferringhi, Penang, MALAYSIA.
- [55] Kanayama, Kimura, Miyazaki and Noguchi, “A Stable Tracking Control Method for a Non-Holonomic Mobile Robot”, IROS’91 Workshop, IEEE, Osaka Japan, Nov 1991,pp. 1236-1241.
- [56] Omid Mohareri ,Rached Dhaouadi , Ahmad B. Rad, Indirect adaptive tracking control of a nonholonomic mobile robot via neural networks, *Neurocomputing* 88 (2012) 54–66.
- [57] W. Dong, W.L. Xu, Adaptive tracking control of uncertain nonholonomic dynamic system, *IEEE Trans. Autom. Control* 46 (2001) 450–454.
- [58] T.Y. Wang, C.C. Tsai, Adaptive trajectory tracking control of a wheeled mobile robot via Lyapunov technique, in: Proceedings of the 30th IEEE Industrial Electronics Conference, 2004, pp. 389–394.
- [59] W. Dong, W. Huo, S.K. Tso, W.L. Xu, Tracking control of uncertain dynamic nonholonomic system and its application to mobile robots, *IEEE Trans. Robot. Automat.* 16 (2000) 870–874.
- [60] P. Coelho, U. Nunes, Path-following control of mobile robots in presence of uncertainties, *IEEE Trans. Robot.* 21 (2005) 252–261.
- [61] G. Oriolo, A.D. Luca, M. Vendittelli, WMR control via dynamic feedback linearization: design, implementation, and experimental validation, *IEEE Trans. Control Syst. Technol.* 10 (2002) 835–852.
- [62] J.M. Yang, J.H. Kim, Sliding mode control for trajectory tracking of nonholonomic wheeled mobile robots, *IEEE Trans. Robot. Autom.* 15 (1999) 578–587.
- [63] H. K. Khalil, *Nonlinear systems*, Prentice Hall, 1996.
- [64] Kennedy and Eberhart., Particle swarm optimization. *IEEE International Conference on Neural Network*, (1995), 1942-1948.
- [65] Zhang Li-ping, Yu Huan-jun, Hu Shang-xu., Optimal choice of parameters for particle swarm optimization. *Journal of Zhejiang University SCIENCE*, (2005), 528-534

- [66] Lingfeng Wang and Chanan Singh., Stochastic combined heat and power dispatch based on multi-objective particle swarm optimization. *Electrical Power and Energy Systems*, 30 (2008). 226–234
- [67] Clerc, M., Kennedy, J., 2002. The particle swarm—explosion, stability, and convergence in a multidimensional complex space. *IEEE Trans. on Evolutionary Computation*, (1):58-73.
- [68] Bansal J.C., Singh P.K., Saraswat M., Verma A., Jadon S.S, Abraham A., (2011) Inertia Weight Strategies in Particle Swarm Optimization, *Third World Congress on Nature and Biologically Inspired Computing*, 978-1-4577-1123-7/11/\$26.00c@2011 IEEE, pp.640-647.
- [69] Motai Y., Jha S.K., Kruse D., Human tracking from a mobile agent: Optical flow and Kalman filter arbitration.
- [70] E.Aguirre, A.Gonzalez,(2000) Fuzzy behaviors for mobile robot navigation: design, coordination and fusion, *International Journal of Approximate Reasoning* Vol.25,pp.255–289.
- [71] J.B.Mbede, W.Weii, Q.S.Zhang, (2001) Fuzzy and recurrent neural network motion control among dynamic obstacles for robot manipulators, *Journal of Intelligent and Robotic Systems*, Vol. 30 , pp.155-177.
- [72] S.J.Kim, W.K.Choi, H.T.Jeon, (2004), Intelligent robot control with personal digital assistants using fuzzy logic and neural network, in: *Knowledge Based Intelligent Information and Engineering Systems*, Pt3, *Proceedings*, Vol.3215, pp.589–595.
- [73] E.Rimon, D.E. Koditschek ,(1992),Exact robot navigation using artificial potential functions , *IEEE Transactions on Robotics and Automation*, Vol.8, pp.501–518.
- [74] J.Borenstein,Y.Koren, (1991), Thevectorfieldhistogram-fast obstacleavoidance formobilerobots,*IEEETransactionsonRoboticsandAutomation*, Vol. 7, pp.278–288
- [75] T. Darrell, G. Gordon, M. Harville, and J. Woodfill. Integrated person tracking using stereo, color, and pattern detection. *Int. Journal of Computer Vision*, 37(2):175–185, 2000.
- [76] S. Feyrer and A. Zell. Robust real-time pursuit of persons with a mobile robot using multisensor fusion. In *Proc. Int. Conf. on Intelligent Autonomous Systems*, pages 710–715, Venice, 2000.
- [77] Choi S., Kim D., (2008),Robust head tracking using 3D ellipsoidal head model in particle filter, *Pattern Recognition*, Vol. 41, pp.2901 – 2915.
- [78] Shen C, Hengel A, Brooks MJ. Visual tracking via efficient kernel discriminate subspace learning. In: *Proceedings of IEEE international conference on image processing*. 2005. p. 590–3.
- [79] Comaniciu D, Ramesh V, Meer P. Kernel-based object tracking. *IEEE Transactionson Pattern Analysis and Machine Intelligence* 2003;25:564–77.

- [80] I.Mikic, M.Trivedi, E.Hunter,P.Cosman, (2003), Human body model acquisition and tracking using voxel data, *International Journal of Computer Vision*, Vol. 53, pp.199–223.
- [81] X.D.Sun, J.Foote, D.Kimber, B.S.Manjunath, (2005), Region of interest extraction and virtual camera control based on panoramic video capturing, *IEEE Transactions on Multimedia*, Vol. 7, pp.981–990.
- [82] D.S.Jang, S.W.Jang, H.I.Choi, (2002), 2D human body tracking with structural Kalman filter, *Pattern Recognition* Vol. 35, pp.2041–2049.
- [83] Petrović E., Ćirić I., Nikolić V.,(2014) , Human tracking by DANI mobile robot based on Extended Kalman filter,, ISSN 1313-4264, Sozopol, Bulgaria, pp.136-142.
- [84] Lia A.,Jinga Z., Hua S.,(2007), Robust observation model for visual tracking in particle filter, *Int. J. Electron. Commun. (AEÜ)*, Vol. 61,pp.186 – 194.
- [85] Saha S., Bambha N., Bhattacharyya S., (2010), Design and implementation of embedded computer vision systems based on particle filters, *Computer Vision and Image Understanding*, Vol.114, pp.1203–1214.
- [86] Petrović E., Durrant D.R., Nikolić V., Leu A., A novel approach to human tracking for Robotic Follower, SAUM 2012, 987-86-6125-072-9, pp.128-131.
- [87] Fox, D., Hightower, J., Liao, L., Schulz, D., & Borriello, G. (2003). Bayesian filtering for location estimation. *IEEE pervasive computing*, 2(3), 24-33.
- [88] Davorin Mikluc, (2016), Procena i izdvajanje parametara ciljeva primenom Partikle filtera u sistemima za praćenje.Doktorska teza.
- [89] Haug, A. J. (2005). A tutorial on Bayesian estimation and tracking techniques applicable to nonlinear and non-Gaussian processes. MITRE Corporation, McLean.
- [90] J. D. Schutter et al, Kalman filters: A Tutorial, Division of Production Engineering, Machine Design and Automation (PMA) Celestijnenlaan 300 B, B-3001 Heverlee, Belgium, October 29, 1999.
- [91] Arulampalam, M. S., Maskell, S., Gordon, N., & Clapp, T. (2002). A tutorial on particle filters for online nonlinear/non-Gaussian Bayesian tracking. *IEEE Transactions on signal processing*, 50(2), 174-188.
- [92] http://automatika.etf.bg.ac.rs/images/FAJLOVI_srpski/predmeti/izborni_kursevi_os/obrada_signala/OS4ASO/materijali/kalman.pdf
- [93] Nikolić V., Ćojbašić Ž., Durrant D.R., Petrović E., Ćirić I.,Matić S., Kalman Filter For Robot Vision-Based Human Tracking, , *Proceedings of 29th DANUBIA-ADRIA Symposium on Advances in Experimental Mechanics*, 978-86-7083-762-1, pp.178-181.
- [94] Welch, G., & Bishop, G. (2006). An introduction to the Kalman filter. Department of Computer Science, University of North Carolina.

- [95] Doucet, Arnaud, Simon Godsill, and Christophe Andrieu. "On sequential Monte Carlo sampling methods for Bayesian filtering." *Statistics and computing* 10.3 (2000): 197-208.
- [96] A. Doucet, J. F. G. de Freitas, and N. J. Gordon, "An introduction to sequential Monte Carlo methods," in *Sequential Monte Carlo Methods in Practice*, A. Doucet, J. F. G. de Freitas, and N. J. Gordon, Eds. New York: Springer-Verlag, 2001.
- [97] Arulampalam, M. S., Maskell, S., Gordon, N., & Clapp, T. (2002). A tutorial on particle filters for online nonlinear/non-Gaussian Bayesian tracking. *IEEE Transactions on signal processing*, 50(2), 174-188.
- [98] Gordon, N., Salmond, D., Smith, A. (1993), Novel Approach to Nonlinear/ Non-gaussian Bayesian State Estimation. *IEE Proceedings Radar and Signal Processing* 140(2), 107-113 .
- [99] J. MacCormick and A. Blake, "A probabilistic exclusion principle for tracking multiple objects," in *Proc. Int. Conf. Comput. Vision*, 1999, pp. 572-578.
- [100] J. Carpenter, P. Clifford, and P. Fearnhead, "Improved particle filter for nonlinear problems," *Proc. Inst. Elect. Eng., Radar, Sonar, Navig.*, 1999.
- [101] D. Crisan, P. Del Moral, and T. J. Lyons, "Non-linear filtering using branching and interacting particle systems," *Markov Processes Related Fields*, vol. 5, no. 3, pp. 293-319, 1999.
- [102] K. Kanazawa, D. Koller, and S. J. Russell, "Stochastic simulation algorithms for dynamic probabilistic networks," in *Proc. Eleventh Annu. Conf. Uncertainty AI*, 1995, pp. 346-351.
- [103] Pitt, Michael K., and Neil Shephard. "Filtering via simulation: Auxiliary particle filters." *Journal of the American statistical association* 94.446 (1999): 590-599.
- [104] Schön, Thomas. "On computational methods for nonlinear estimation." (2003). *Linköping Studies in Science and Technology. Licentiate Thesis No. 1047*, Linköping University, Linköping, Sweden, October 200.
- [105] Andrieu, C., and S. J. Godsill. "A particle filter for model based audio source separation." *Proceedings of the International Workshop on Independent Component Analysis and Blind Signal Separation (ICA 2000)*. 2000.
- [106] Särkkä, S. (2013). *Bayesian filtering and smoothing (Vol. 3)*. Cambridge University Press.
- [107] Haug, A. J. (2005). A tutorial on Bayesian estimation and tracking techniques applicable to nonlinear and non-Gaussian processes. MITRE Corporation, McLean.
- [108] Salmond, D., & Gordon, N. (2005). *An introduction to particle filters*.
- [109] Montesano, L. (2006). *Detection and tracking of moving objects from a mobile platform. application to navigation and multi-robot localization (Doctoral dissertation, PhD thesis, Universidad de Zaragoza)*.

- [110] Bumblebee product description [online] Available at: <<http://www.ptgrey.com/products/bbxb3>> [Accessed August 2012].
- [111] Ristić-Durrant D., Gao G., Leu A., Low-level sensor fusion-based human tracking for mobile robot, Facta Universitatis, Series: Automatic Control and Robotics, 2015.
- [112] Bosch PLR 50 product description [online] Available at: <<http://www.bosch-do-it.de/boptocs2-de/Heimwerker/Werkzeuge/DE/de/hw/Entfernungsmesser/95299/PLR+50/24116/3165140532518/index.htm>> [Accessed August 2012].
- [113] <http://www.ni.com/datasheet/pdf/en/ds-217>
- [114] <http://www.ni.com/white-paper/11564/en/>
- [115] <http://www.acrosser.com/Products/Embedded-Computer/Fanless-Embedded-Systems/AES-HM76Z1FL/Intel-Core-i3%EF%BC%8Fi7-AES-HM76Z1FL.html>
- [116] M. Bennewitz, G. Cielniak, and W. Burgard, “Utilizing learned motion patterns to robustly track persons,” in Proc. of Joint IEEE Int. Workshop on VS-PETS, Nice, France, 2003, pp. 102–109.
- [117] Kim, S. J., Park, J. W., & Lee, J. (2005). Implementation of tracking and capturing a moving object using a mobile robot. *International Journal of Control Automation and Systems*, 3(3), 444.
- [118] D. Schulz, W. Burgard, D. Fox, and A. B. Cremers, “People tracking with mobile robots using sample-based joint probabilistic data association filters,” *Int. Journal of Robotics Research*, vol. 22, no. 2, pp. 99–116, 2003.
- [119] Bellotto, N., & Hu, H. (2007, August). People tracking and identification with a mobile robot. In *2007 International Conference on Mechatronics and Automation* (pp. 3565-3570). IEEE.
- [120] Bellotto, N., & Hu, H. (2007, August). People tracking with a mobile robot: A comparison of Kalman and particle filters. In *Proc. of the 13th IASTED Int. Conf. on Robotics and Applications* (pp. 388-393).
- [121] Karan, B. (2015). Calibration of Kinect-type RGB-D sensors for robotic applications. *FME Transactions*, 43(1), 47-54.
- [122] Chen, C., Jafari, R., & Kehtarnavaz, N. (2015). Improving human action recognition using fusion of depth camera and inertial sensors. *IEEE Transactions on Human-Machine Systems*, 45(1), 51-61.
- [123] Shilpashree S ., D.J.Ravi, Jagadeesh B, Human Tracking and Action Recognition Based on k-Nearest Neighbor Classifier, *International Journal of Engineering Research in Electronic and Communication Engineering (IJERECE)* Vol 3, Issue 5, May 2016.

- [124] Anguita, D., Ghio, A., Oneto, L., Parra, X., & Reyes-Ortiz, J. L. (2012, December). Human activity recognition on smartphones using a multiclass hardware-friendly support vector machine. In *International Workshop on Ambient Assisted Living* (pp. 216-223). Springer Berlin Heidelberg.
- [125] Simon, C., Meessen, J., & De Vleeschouwer, C. (2010). Visual event recognition using decision trees. *Multimedia Tools and Applications*, 50(1), 95-121.
- [126] Preece, S.J., Goulermas, J.Y., Kenney, L.P., Howard, D. (2009). A comparison of feature extraction methods for the classification of dynamic activities from accelerometer data. *IEEE Transactions on Biomedical Engineering*, 56 (3), 871-879.
- [127] Oniga, S., & József, S. (2015). Optimal Recognition Method of Human Activities Using Artificial Neural Networks. *Measurement Science Review*, 15(6), 323-327.
- [128] Benyacoub, B., ElBernoussi, S., Zoglat, A., & Ismail, E. M. (2014). Classification with Hidden Markov Model. *Applied Mathematical Sciences*, 8(50), 2483-2496.
- [129] Ristić-Durrant D., Gräser A., Cognitive Robotics: Towards symbiotic human-robot relationship, XI International SAUM Conference on Systems, Automatic Control and Measurements Niš, Serbia, November 14th-16th, 2012.
- [130] Gao, L., Bourke, A.K., Nelson, J. (2011). A system for activity recognition using multisensor fusion. In *Annual International Conference of the IEEE -Engineering in Medicine and Biology Society (EMBC 2011)*, Boston, MA. IEEE, 7869-7872.
- [131] Magnanimo, V., Saveriano, M., Rossi, S., & Lee, D. (2014, August). A Bayesian approach for task recognition and future human activity prediction. In *The 23rd IEEE International Symposium on Robot and Human Interactive Communication* (pp. 726-731). IEEE.
- [132] Ćirić Ivan, Inteligentno upravljanje mobilnim robotima na osnovu neuro-fazi-genetskog prepoznavanja objekata i praćenja ljudi u robotskoj viziji, (2016), PhD thesis, Presented to the Faculty of the Mechanical Engineering University of Niš.
- [133] Bellotto, N. (2012). Robot control based on qualitative representation of human trajectories.
- [134] Cohn, A. G., and Renz, J. 2008. Chapter 13 Qualitative Spatial Representation and Reasoning. In van Harmelen, F.; Lifschitz, V.; and Porter, B., eds., *Handbook of Knowledge Representation*, volume 3. Elsevier. 551–596.
- [135] Zhang, G. P. (2000). Neural networks for classification: a survey. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, 30(4), 451-462.
- [136] <http://www.ni.com/pdf/manuals/370420b.pdf>

DODATAK A

Funkcija 1: Kinematika robota

```
function z=Kinematika(v,omega,theta)
S=[cos(theta) 0; sin(theta) 0; 0 1];
u=[v;omega];
qt=S*u;
z=[qt(1);qt(2);qt(3)];
```

Funkcija 2: Dinamika robota

```
function y=Dinamika(TR,TL,v,omega,theta)
m=3.6;           % Masa robota
r=0.05;         % Poluprecnik tockova
d=0.05;         % Rastojanje izmedju centra mase i ose tocka u pravcu x-ose
rc = 0.2025;    % Rastojanje tocka od ose simetrije robota u pravcu y-ose
Ic=0.09;        % Moment inercije centra mase robota
I=Ic+m*d^2;
M=[m 0;0 I];
C=[0 -m*d*omega; m*d*omega 0];
B=[1/r 1/r;rc/r -rc/r];
T=[TR;TL];
u=[v;omega];
ut=inv(M)*(B*T-C*u);
y=[ut(1); ut(2)];
```

Funkcija 3: Putanja čoveka-kvadrat

```
function u = putanja_kvadrat(t)

% vreme trajanja trajektorije
T1=2;
T2=4;
T3=6;
T4=8;

if t<=T1

x=t;           %Izracunavanje x(t)
```



```

y=0;          %Izracunavanje y(t)na osnovu putanje u xy ravni
vx=1;        %Izracunavanje brzina vx and vy ( prvi izvod)
vy=0;
v=(vx^2+vy^2)^0.5; % Izracunavanje brzine v

elseif t<=T2

x=2;         %Izracunavanje x(t)
y=t-2;       %Izracunavanje y(t)na osnovu putanje u xy ravni
vx=0;        %Izracunavanje brzina vx and vy ( prvi izvod)
vy=1;
v=(vx^2+vy^2)^0.5; % Izracunavanje brzine v

elseif t<=T3

x=-t+6;      %Izracunavanje x(t)
y=2;         %Izracunavanje y(t)na osnovu putanje u xy ravni
vx=1;        %Izracunavanje brzina vx and vy ( prvi izvod)
vy=0;
v=(vx^2+vy^2)^0.5; % Izracunavanje brzine v

else t<=T4;

x=0;         %Izracunavanje x(t)
y=-t+8;     %Izracunavanje y(t)na osnovu putanje u xy ravni
vx=0;        %Izracunavanje brzina vx and vy ( prvi izvod)
vy=1;
v=(vx^2+vy^2)^0.5; % Izracunavanje brzine v
end
u=[x,y,v];
end

```

DODATAK B

Funkcija 1: Linearni Kalmanov filter

```

% brisanje svega
clear all
close all
clc

```

```

data = xlsread('Test_linear_KF.xls');% učitavanje detektovane pozicije
frameNo = data(:,1); % broj frejmova
dataX = data(:,2); % pozicija Xc
dataZ = data(:,3); % pozicija Zc
measurement.exist=data(:,4) ; % status pracenja
Y=[dataX,dataZ]'; % vektor merenja

% Inicijalizacija Kalmanovog filtera

x = [0;3;0;0];      % inicijalno stanje
F = [1 0 0 0;0 1 0 0;0 0 1 0;0 0 0 1]; % prelazna matrica stanja
Q = eye(4)*1e1;     % kovarijaciona matrica procesa
H = [ 1 0 0 0;0 1 0 0]; % matrica merenja
R = eye(2)*1e2;     % kovarijaciona matrica merenja
P = eye(4)*1e-2;    %inicijalna kovarijaciona matrica greske estimacije
T=length(Y);       % broj iteracija

X_est = []; %inicijalni niz estimiranog stanja x
Z_est = []; %inicijalni niz estimiranog stanja z

for n=1:T

    z=Y(:,n)

    %korak 1: PREDIKCIJA
    [x,P]= kalman_predict(x,P,F,Q);

% korak 2: KOREKCIJA

    %%% Ukoliko vizija ne daje podatke vrednost promenljive
    measurement.exist je 0 i ukoliko nemamo podatke sa senzora vizije ne
    vrsimo korekciju

if measurement.exist (n,1)==1

    [x,P]= kalman_update(x,P,z,H,R);

else

    [x,P]

```

```

end

    est_x=x(1,1); % estimirana vrednost stanja X
est_z=x(2,1); % estimirana vrednost stanja Z

% cuvanje podataka u nizovima za kasnije crtanje
    X_est = [X_est est_x];
    Z_est = [Z_est est_z];

end

% crtanje

figure (1)
plot(Y(1,:), '-r', 'LineWidth', 2)
hold on
plot(X_est, 'b'), title ('Xc pozicija coveka')
ylabel ('Xc(m)')
xlabel ('Frame')
legend('Pozicija dobijena stereo kamerom', 'Pozicija dobijena Kalmanovim
filterom')
grid on
hold off
figure(2)
plot(Y(2,:), '-r', 'LineWidth', 2 )
hold on
plot(Z_est, 'b'), title ('Zc pozicija coveka')
ylabel ('Zc(m)')
xlabel ('Frame')
legend('Pozicija dobijena stereo kamerom', 'Pozicija dobijena Kalmanovim
filterom')
grid on
hold off

.....

function [x,P] = kalman_predict(x,P,F,Q)
    x = F*x;      %prediktovano stanje
    P = F*P*F' + Q; %prediktovana kovarijaciona matrica greske estimacije
end

```

```

.....

function [x,P] = kalman_update(x,P,y,H,R)
z = y - H*x;          %greska merenja
S = H*P*H' + R; %kovarijaciona matrica greške merenja
K = P*H'*inv(S);%optimalno Kalmanovo pojačanje
x = x + K*z;          % azurirano stanje
P = (eye(size(x,1))-K*H)*P;%azurirana kovarijaciona matrica greske estimacije
end

```

Funkcija 2: Prošireni Kalmanov filter

```

% brisanje svega
clear all
close all
clc

data = xlsread('Test_nonlinear.xls');% učitavanje detektovane pozicije
frameNo = data(:,1); % broj frejmova
dataD = data(:,2); % rastojanje D
dataFi = data(:,3); % ugao fi
measurement.exist=data(:,4) ; % status pracenja
Y=[dataD,dataFi]'; % vektor merenja

% Inicijalizacija prosirenog Kalmanovog filtera

x = [0;4;0;0]; % inicijalno stanje
T=length(Y); % broj iteracija
Q = eye(4)*1e-2; % kovarijaciona matrica procesa
P = eye(4)*1e-2; %inicijalna kovarijaciona matrica greske estimacije
R=[0.01 0;0 0.001];% kovarijaciona matrica merenja

X_est = []; %inicijalni niz estimiranog stanja x
Z_est = []; %inicijalni niz estimiranog stanja z
D_est = []; %inicijalni niz estimiranog stanja d
F_est = []; %inicijalni niz estimiranog stanja fi

for i=1:T
    y=Y(:,i);

```

```

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%korak 1: PREDIKCIJA
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
        [F]=Jacobianf(x);           % racunanje Jakobijana za funkciju f
        [x,P]= kalman_predict(x,P,F,Q); % predikcija stanja

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% korak 2: KOREKCIJA
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
        %%% Ukoliko vizija ne daje podatke vrednost promenljive
        measurement.exist je 0 i ukoliko nemamo podatke sa senzora vizije ne
        vrsimo korekciju
if measurement.exist (n,1)==1

        [H]=Jacobianh(x); % racunanje Jakobijana za funkciju h
[x,P]= kalman_update(x,P,y,H,R); % korekcija

else

        [x,P]

end

        est_x=x(1,1); % estimirana vrednost stanja X
est_z=x(2,1); % estimirana vrednost stanja Z
        est_d=sqrt(x(1,1)^2+x(2,1)^2); % estimirana vrednost distance d
        est_f=atan(x(1,1)/x(2,1)); % estimirana vrednost ugla fi

% cuvanje podataka u nizovima za kasnije crtanje
        X_est = [X_est est_x];
        Z_est = [Z_est est_z];
D_est = [D_est est_d];
        F_est = [F_est est_f];
end

.....

function [x,P] = kalman_predict(x,P,F,Q)
        x = F*x; %prediktovano stanje

```

```

    P = F*P*F' + Q; %prediktovana kovarijaciona matrica greske estimacije
end

```

```

.....

function [x,P] = kalman_update(x,P,y,H,R)

    yh=[sqrt(x(1,1)^2+x(2,1)^2);atan(x(1,1)/x(2,1))];
    z = y - yh; %greške merenja
    S = H*P*H' + R; %kovarijaciona matrica greške merenja
    K = P*H'*inv(S);%optimalno Kalmanovo pojacanje
    x = x + K*z;      % azurirano stanje
    P = (eye(size(x,1))-K*H)*P;%azurirana kovarijaciona matrica greske
end

```

```

.....

function [F]= Jacobian(x)
F=[ 1 0 1 0;
    0 1 0 1;
    0 0 1 0;
    0 0 0 1]
end

```

```

.....

function [H]= Jacobian(x)
H=[ x(1,1)/sqrt(x(1,1)^2+x(2,1)),x(2,1)/sqrt(x(1,1)^2+x(2,1)) 0 0;
    x(2,1)/(x(1,1)^2+x(2,1)) x(1,1)/sqrt(x(1,1)^2+x(2,1)) 0 0];
end

```

Funkcija 3: Partikle filter

```

% brisanje svega
clear all
close all
clc

data = xlsread('Test_nonlinear.xls');
frameNo = data(:,1);
dataD = data(:,2);

```

```

dataFi = data(:,3);
measurement.exist=data(:,4) ; % status pracenja
Y=[dataD,dataFi]';

% Inicijalizacija varijabli
x0 = [0;4;0;0]; % inicijalno stanje
T=length(Y); % number of interation
R=[0.01 0;0 0.001]; % kovarijaciona matrica merenja
Q = eye(4)*1e-2; % kovarijaciona matrica procesa
P = eye(4)*1e-2; % inicijalna kovarijaciona matrica greske estimacije
m=length(x0); % broj stanja
N=2000; % broj partikal(cestica). Sto je veci broj, bolja je
aprokcimacija, ali je racunski zahtevnije.

est=[]; % i%inicijalni niz estimiranog stanja

for i=1:N
p(:,i)=x0+sqrtm(P)*randn(4,1); % inicijalizacija partikla
end

for t = 1:T

% Partikle filter

for i = 1:N
y=Y(:,t);

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% korak 1: PREDIKCIJA
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

xp(:,i) = f(p(:,i))+sqrtm(Q)*randn(4,1); % predikcija na osnovu modela

yp(:,i) = h(xp(:,i))+sqrtm(R)*randn(2,1); % u ovom koraku vrsimo
azuriranje opservacije sa novim setom partikla

% Napomena, nemamo puno stanje opservacije, nego samo merenja trenutne pozicije
i prema tome merenje je zasnovano samo na gresci pozicije

```

```

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% korak 2: KOREKCIJA
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

    if measurement.exist (n,1)==1

        %%% Ukoliko vizija ne daje podatke vrednost promenljive
        measurement.exist je 0 i ukoliko nemamo podatke sa senzora vizije ne
        vrsimo korekciju

        d = y - yp(:,i); % razlika izmedju trenutne pozicije (merenja) i
        opservacije (predvidjene pozicije)

        q(i)=1/(2*pi*sqrt(det(R)))*exp(-0.5*d'*inv(R)*d);% dodeljujemo
        tezinu svakoj partikli(cestici)

    end
end

if measurement.exist (n,1)==1

    q=q./sum(q); % normalizujemo tezine

    x_est=q*xp';      % estimirana vrednost stanja
    x_est=x_est';

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% korak 3: RESAMPLOVANJE
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

Neff=1/sum(q.^2); % broj efektivnih cestica

if Neff<3/4*N

    for k=1:m
        p(k,:) = re_sample(xp(k,:),q); % resamplovanje
    end
else
    for k=1:m
        p(k,:) =q.*xp(k,:);
    end
end
end

```



```

else
    x_est=mean(xp,2);    %estimirana vrednost stanja kada nemamo merenje
end

```

```

    est = [est x_est]; %cuvanje podataka u nizovima za kasnije crtanje

```

```

end

```

```

% Podaci za crtanje

```

```

X_est =est(1,:);

```

```

Z_est =est(2,:);

```

```

.....

```

```

function xp = f(x) % prelazna jednacina stanja

```

```

F=[1 0 1 0;0 1 0 1;0 0 1 0;0 0 0 1];

```

```

xp = F*x;

```

```

end

```

```

.....

```

```

function y = h(x) % nelinearna opservaciona funkcija

```

```

y1 = sqrt(x(1,1).^2+ x(2,1).^2);

```

```

y2=atan(x(1,1)./x(2,1));

```

```

y=[y1;y2];

```

```

end

```

```

.....

```

```

function p = re_sample(xp,q) % resamplovanje (thanks to Nino's Matrix Lab)

```

```

cdf = cumsum(q); % kumulativna suma

```

```

diff = cdf'*ones(1,length(q)) - ones(length(q),1)*rand(1,length(q));

```

```

diff = (diff <= 0) * 2 + diff;

```

```

[~, idx] = min(diff);

```

```

p = xp(idx);

```

```

end

```

Funkcija 4: Real time Kalman filter

```

function [est_x,est_z,P]=kalman(tracking_status,xHC,zHC,DM_x,DM_z,DM_P)

```

```

% Inicijalizacija Kalmanovog filtera

```

```

x = [DM_x;DM_z;0;0];          % stanje u trenutki k-1
F = [1 0 1 0;0 1 0 1;0 0 1 0;0 0 0 1]; % prelazna matrica stanja
Q = eye(4)*1e-4;             % kovarijaciona matrica procesa
H = [ 1 0 0 0;0 1 0 0]; % matrica merenja
R = eye(2)*10;               % kovarijaciona matrica merenja
P = DM_P;                     % kovarijaciona matrica greske estimacije u trenutki k-1
z=[xHC;zHC];

%korak 1: PREDIKCIJA
[x,P]= kalman_predict(x,P,F,Q);

% korak 2: KOREKCIJA

    %%% Ukoliko vizija ne daje podatke vrednost promenljive
    tracking_status je 0 i ukoliko nemamo podatke sa senzora vizije ne
    vrsimo korekciju

if tracking_status==1

    [x,P]= kalman_update(x,P,z,H,R);

else

    [x,P]

end

    est_x=x(1,1); % estimirana vrednost stanja X
    est_z=x(2,1); % estimirana vrednost stanja Z
P;
end

function [x,P] = kalman_predict(x,P,F,Q)
    x = F*x;          %prediktovano stanje
    P = F*P*F' + Q; %prediktovana kovarijaciona matrica greske estimacije
end

function [x,P] = kalman_update(x,P,y,H,R)
z = y - H*x;          %greska merenja
S = H*P*H' + R;      %kovarijaciona matrica greške merenja
K = P*H'*inv(S);     %optimalno Kalmanovo pojačanje
x = x + K*z;         % azurirano stanje
P = (eye(size(x,1))-K*H)*P; %azurirana kovarijaciona matrica greske estimacije
end

```