University of Belgrade
School of Electrical Engineering

Zaviša Gordić

Detection of Interaction Forces in Industrial Robotics

Doctoral Dissertation

Belgrade, 2022.

Универзитет у Београду
Електротехнички факултет

Завиша Гордић

Детекција сила интеракције у индустријској роботици

Докторска дисертација

Београд, 2022.

Doctoral Dissertation Supervisor

Dr. **Kosta Jovanović**
Associate Professor
University of Belgrade,
School of Electrical Engineering

Doctoral Dissertation Committee Members

Dr. **Željko Đurović**
Full Professor
University of Belgrade,
School of Electrical Engineering

Dr. **Aleksandar Rodić**
Scientific Advisor
University of Belgrade,
Institute Mihajlo Pupin

Dr. **Zoran Miljković**
Full Professor
University of Belgrade,
Faculty of Mechanical Engineering

Dr. **Nenad Jovičić**
Associate Professor
University of Belgrade,
School of Electrical Engineering

Defence date:

ACKNOWLEDGEMENTS

I would like to start by thanking my mentor, Dr Kosta Jovanović for his support, understanding and encouragement during PhD studies. Working with him has enriched my professional experience and opened numerous possibilities and perspectives which have also contributed to my personal development.

I extend my thanks to the members of the Committee for their constructive suggestions which have contributed to the final version of the dissertation.

During my studies, I was privileged to have the opportunity to learn from many inspiring and admirable members of teaching staff, both in professional and personal aspect. Among them, I would especially like to thank professor Željko Đurović who mentored my bachelor and master thesis and provided advice and inspiration throughout entire studies, professor Veljko Potkonjak who was my mentor at the beginning of PhD studies, and whose lessons I will always remember. Special and kind gratitude I owe to professor Aleksandar Rodic for his inspiring enthusiasm as well as to professor Tomislav Šekara with whose selfless support and guidance I wrote my first scientific papers.

Studies and the work on the dissertation were a great deal more interesting and enjoyable experience thanks to my office colleagues, especially Branko Lukić, Nikola Knežević, Maja Trumić and Vladimir Petrović. Time spent with them, their moral and practical support will always be remembered.

I would also like to give thanks to Claudio Ongaro, Fabio Greco and Branko Nikolić who helped direct my research attention towards industrial robotics, and with whose support I made my first scientific contributions in the field.

Undoubtably the biggest debt of gratitude for immense and unreserved support and encouragement on every step during studies I owe to my family, especially to my wife Gordana and mother Svetlana. Thank you for your patience and understanding.

Dissertation title: Detection of Interaction Forces in Industrial Robotics

Abstract: With Industry 4.0 becoming a reality and Industry 5.0 emerging on the horizon, the need for seamless integration, shared workspace and interoperability of production entities is ever increasing. To aid in this transition, this thesis presents approaches intended to allow the evolution of industrial robots by enabling them to detect and interpret interactions with their surroundings. The detection of interaction forces is based on non-model-based algorithms due to their inherent ability to include all aspects of the behaviours of the robot as well as to capture the contact task-specific forces and dynamics. To detect interactions, the reference sequence recorded during an exemplary task execution cycle is compared with measurements from the robot while it is performing its repetitive task. The thesis presents several different approaches to detection of collisions and interactions in general intended for the implementation on industrial robots with closed control architecture. To overcome implementation issues, the modified Dynamic Time Warping (mDTW) method, as one of the key presented contributions, enables optimal matching of compared signals. The mDTW enables comparing a signal with the most similar section of the other signal. Partial matching also enables online application of time warping principles and reduces the time and computation resources needed to perform matching. The developed and presented algorithms for automatic calculation of kinematic parameters of the robot and its end-effector enable further evolution of the mDTW in into its kinematically augmented version - KA-mDTW, extending the interaction's detection algorithm's application domain. Furthermore, it enables the inclusion of unmodeled task dynamics or a robot's end-effector into algorithms for collision detection or general understanding of a robot's operation context. The presented algorithms and conclusions are supported and validated by the experimental testing on industrial robots.

Keywords: Industrial robotics; Interaction detection, Physical interaction, Dynamic Time Warping, Industry 4.0

Scientific field: Electrical and Computer Engineering

Scientific subfield: Robotics and Control Systems

Наслов дисертације: Детекција сила интеракције у индустријској роботици

Сажетак: Услед Индустрији 4.0 која постаје стварност и Индустрије 5.0 која се јавља на видику, све је јача потреба за олакшаном интеграцијом, дељеним радним простором и међуповезаношћу производних чинилаца. Као одговор на те потребе, ова дисертација представља приступе намењене еволуцији индустријских робота тако што им омогућава да опажају и тумаче интеракцију са својим окружењем. Детекција сила интеракције је заснована на алгоритмима који не захтевају модел робота због њихове особине да посредно укључе све чиниоце понашања робота и специфичних сила везаних за динамику интеракције. Детекција интеракције се постиже поређењем референтних вредности снимљених током репрезентативног покрета са мерењима добијених од робота током вршења свог поновљивог задатка. Дисертација представља неколико различитих приступа намењених примени на индустријским роботима са затвореном управљачком архитектуром а који се могу користити ради детекције судара и интеракције уопште. Како би се превазишли изазови везани за примену, један од главних доприноса, развијени modified Dynamic Time Warping (mDTW) метод, омогућава оптимално упаривање поређених сигнала. Представљени mDTW метод омогућава упаривање једног сигнала са најсличнијим делом другог сигнала. Упаривање делова омогућава примену Time Warping-а у реалном времену и смањује време потребно за обраду података. Развијени и представљени алгоритми за аутоматско рачунање кинематичких параметара робота и завршног уређаја омогућавају даљу еволуцију mDTW-а у његову кинематички допуњену верзију - KA-mDTW, чиме се проширује поље примене алгоритма за детекцију интеракције. Овим се додатно омогућава укључивање немоделоване динамике роботског задатка или завршног уређаја у циљу детекције судара или општег разумевања оперативног контекста. Представљени алгоритми и закључци су потврђени и проверени експерименталним тестирањем на индустријским роботима.

.

Кључне речи: Индустријска роботика; Детекција интеракције; Физичка интеракција; Dynamic Time Warping, Индустрија 4.0

Научна област: Електротехника и рачунарство

Ужа научна област: Роботика и управљање системима

CONTENTS

1. INTRODUCTION

Tendencies in modern day robotics demand a transition to a more flexible, adaptable, and interconnected event-driven environment. Availability of information in form of process-related measurements and operation context is of crucial importance for a successful integration and optimized control of all processing and management entities.

These transitions reflect not only on a control and sensing level, but also on the physical workspace of the production resources. Ever increasing requirements for sharing workspace between industrial robots and other production entities open numerous possibilities in framework of Industry 4.0 and slowly emerging Industry 5.0, but also a multitude of related challenges.

As the backbone of the Third industrial revolution, industrial robots were developed to achieve best performance in highly structured and supervised environments. They were designed to be reliable, mechanically robust, and fast machines with high positioning repeatability which work in closed work cells without interference of humans or other sources of unpredictability. Absence of potential external disturbances meant that their sensors, actuators, and control algorithms were focused on achieving best performance without considering their surroundings.

The evolution of the architecture of interaction [1] between the robotized production cells and other manufacturing entities including mobile robots, human workers and manned vehicles results in violation of the structured environment which the industrial robots were intended for. Addressing these violations inadequately may lead to numerous types of erroneous states and events such as incorrect or suboptimal task execution [2], and potentially to more severe consequences such as damage of production resources, or even injury of humans [3], [4], [5].

Collaborative robots were conceived as the answer to the emerging requirements from the standpoint of safety and sharing workspace with humans, and there are certainly fields where their implementation is the key enabling factor for a successful robotization. However, their performance and robustness are often not the answer for environments where high efficiency and repeatability are required, and where human presence or intervention is not frequent. Moreover, although there are fields where new developments in automation have enabled successful robotization for the first time, in most cases they are introduced as improvements into already existing robotized systems. In such systems and environments, industrial robots are by far the most dominant robot type [6], [7] , and projections for the foreseeable future predict that the demand for them will only increase.

No matter how revolutionary some shifts in automation paradigm are, their influence on the manufacturing components is that of evolution. The same is true for the industrial robots. One important direction in which industrial robots need to evolve to respond to these shifts is to appropriately detect, interpret, react, and share information about the physical interactions they encounter during operation.

1.1.Motivation

The motivation for the work presented in this thesis is to develop algorithms that can unlock the underused potential of industrial robots and their readily available sensors to facilitate their transition into Industry 4.0 and beyond. Detection, interpretation, and reaction to the forces that occur during physical interaction of the robot and its surrounding is one of the prime examples and most impactful fields in which the contribution to this transition can be made.

Collision detection and reaction on different levels [8], [9] is one of the primary and essential aspects that need to be addressed in shared workspace, especially during physical human-robot interaction (pHRI). The research presented in the thesis is focused on the co-existence [8], as the innermost level of any pHRI that is the last instance preventing the human injury if all other levels fail. Besides reliable detection, the reaction time in case of collisions is of the greatest importance from the perspective of human safety. Applicability is another important aspect since collision detection algorithms only contribute to human safety and protection of production resources if they are implemented.

Identification of intentional interaction while performing contact tasks is another important aspect of robotized production. It enables interpretation of the forces that occur during physical contact to determine whether they indicate correct task execution or some erroneous state, such as misalignment with the work object, dropped load, irregular behaviour of the robot or its tool. Information about physical interaction can further be used for predictive maintenance or providing context information regarding operation state of the robot and its equipment.

The main attention of the research is directed towards industrial robots with closed controller architecture [10], [11], [12], [13], [14] since they are the most common and most challenging type of industrial robot controllers. Closed architecture of industrial robots' controllers imposes certain limitations on the availability of information and levels of control possible to attain. These controllers do not allow access or insight to the inner control loops nor the parameters of the model of robot for which they are optimally designed.

The algorithms presented in this thesis rely exclusively on readily available signals possible to attain from closed architecture controllers at a robot joint level. This makes the presented algorithms universally applicable to robots of different configurations and generations, enabling their evolution and facilitating their seamless integration into modern and future production environments.

1.2. Contributions

The research within the thesis has been devoted to the development of algorithms that will enable the industrial robots to detect interactions and better integrate with their surroundings. The work presented in this thesis summarizes and heavily relies on the related work presented throughout the studies. The related publications by the author are:

International journals:

[IJ1] **Z. Gordić,** K. Jovanović: A Framework for Inclusion of Unmodelled Contact Tasks Dynamics in Industrial Robotics, Sensors (ISSN: 1424-8220), Vol. 22, No. 19, pp. 1-19, Nov, 2022

[IJ2] **Z. Gordić**, K. Jovanović: Collision Detection on Industrial Robots in Repetitive Tasks Using Modified Dynamic Time Warping, Robotica (ISSN: 0263-5747), Vol. 38, No. 10, pp. 1717-1736, 2020

National journals:

[DJ1] **Z. Gordić**, C. Ongaro, Calibration of Robot Tool Centre Pointusing Camera-based System, Serbian Journal Of Electrical Engineering, Vol. 13, No. 1, pp. 9-20, 2016.

International conferences:

[IC1] **Z. Gordić**, K. Jovanović, Influence of Unmodelled External Forces on the Quality of Collision Detection, Advances in Service and Industrial Robotics. RAAD 2019. Advances in Intelligent Systems and Computing, pp. 319-328, Springer, Cham, Kaiserslautern, Germany, Jun, 2019.

[IC2] **Z. Gordić**, K. Jovanović, Fully Integrated Torque-Based Collision Detection in Periodic Tasks for Industrial Robots with Closed Control Architecture, Advances in Service and Industrial Robotics Proceedings of the 27th International Conference on Robotics in Alpe-Adria Danube Region (RAAD 2018), pp. 71-81, Springer, Greece, Jun, 2018.

[IC3] **Z. Gordić**, C. Ongaro, Development and Implementation of Orthogonal Planes Images Method, Advances in Intelligent Systems and Computing (Proc. 25th IFTOMM/IEEE International Conference on Robotics in Alpe-Adria- Danube Region – RAAD 2016), pp. 105-115, Springer, Serbia, 2017.

National conferences:

[DC1] **Z. Gordić**, K. Jovanović, Identifying Unmodelled Dynamics in Contact Tasks in Industrial Robotics, Proceedings of the 7th International Conference on Electrical, Electronic and Computing Engineering, IcETRAN 2020, pp. 695-700, Jun, 2020.

[DC2] Z. Gordić, K. Jovanović: Collision Detection on Industrial Robot using Dynamic Time Warping, Proceedings of the 5th International Conference on Electrical, Electronic and Computing Engineering (IcETRAN 2018), pp. 1039-1043, Društvo ETRAN, Serbia, Jun, 2018.

[DC3] **Z. Gordić**, K. Jovanović, Partial Pose Measurements for Identification of Denavit-Hartenberg Parameters of an Industrial Robot, Proceedings of the 4th International Conference on Electrical, Electronic and Computing Engineering (IcETRAN 2017), pp. ROI1.6.1-ROI1.6.4, Друштво за ЕТРАН, Kladovo, Serbia, Jun, 2017.

[DC4] N. Knežević, K. Jovanović, **Z. Gordić**, V. Potkonjak, M. Majstorović, Hazard Identification, Risk Assessment and Safety Integration for Flexible Robotic Cell, Proceedings of the 4th International Conference on Electrical, Electronic and Computing Engineering (IcETRAN 2017), Друштво за ЕТРАН, Jun, 2017.

[DC5] **Z. Gordić**, V. Potkonjak, Overview of Methods for Robotic Manipulators Calibration, Proceedings of the 3rd International Conference on Electrical, Electronic and Computing Engineering (IcETRAN 2016), pp. ROI2.6.1-ROI2.6.4, Društvo ETRAN, Serbia, 2016.

[DC6] **Z. Gordić**, C. Ongaro, Robot Tool Centre Point Calibration using Analysis of Images from Orthogonal Planes, Proceedings of 2nd International Conference on Electrical, Electronic and Computing Engineering, pp. ROI4.6.1-ROI4.6.5, Jun, 2015.

The contribution of the research presented in the thesis can be divided into two principal categories:

1. Detection of interactions and safety:
    a. Detection of interactions based on the principles and modifications of Dynamic Time Warping: IJ1, IJ2, IC1, DC1, DC2
    b. Collision detection and safety for application on industrial robots with closed control architecture: IJ1, IJ2, IC1, IC2, DC1, DC2, DC4
2. Identification of robot kinematics parameters:
    a. Identification of kinematic parameters of the robot manipulator based on partial pose measurements of position of a point of interest at the robot's end effector: DC3, DC5
    b. Identification of kinematic parameters of the robot end-effector based on the analysis of images obtained from cameras in two orthogonal planes. DJ1, IC3, DC5, DC6

1.3. Thesis overview

The thesis is divided into five Chapters.

The First Chapter introduces the reader into the topic, motivation, contributions, and the structure of the thesis.

The Second Chapter is dedicated to the overview of methods for detection of interactions. It is divided into sections dedicated to the different approaches to detection and interpretation of external forces. It discusses advantages and disadvantages of the model-based and non-model-based approaches in general and offers reasoning behind opting for the latter approach in the thesis. The Chapter also offers initial considerations regarding a hybrid approach to interaction detection, combining prospects of the two main groups of approaches.

This Chapter also presents the analysis of signals that are readily available on robots with closed control architecture. The discussion focuses on the type, availability and statistical properties od available signals and their possibility for implementation in the field of interaction detection.

The Third Chapter presents developed algorithms for non-model-based interaction detection for tasks which repeat in cycles in identical way. To this end, two approaches are presented and discussed for implementation on industrial robots with closed control architecture.

With aim of reducing reaction time to possible collisions, the first approach is intended for integration on the controller of the robot itself. The algorithm considers limitations imposed by the closed control architecture and proposes applicable solutions for efficient collision detection.

Aiming to extend the possibilities and increase reliability and sensitivity of detection, the second approach relies on implementation of more complex signal-matching algorithms enabled by the permanent connection with PC. This approach proposes an effective modification of elastic similarity measures to overcome the sampling issues and enable real-time implementation of performant collision detection.

Following the discussion of results obtained from the experimental setup, the proposed approach for collision detection is discussed also in scope of a hybrid interaction detection in which it would be used to detect intentional interactions.

The Fourth Chapter presents an innovative augmentation of the methods used in the Third Chapter, enabling matching signals from similar tasks. The alteration which enables the extension of the application field is based on the identification of kinematic parameters of the robot. With a goal of identification of parameters of the robot manipulator itself and its end effector, two innovative approaches are presented and discussed in terms of their reliability and applicability.

The algorithm for detection of the parameters of the robot itself is based on obtaining partial-pose measurements of a point of interest on at the robot's end effector and can be used for forming kinematic model of robot without limits to its joint configuration.

The algorithm for determining parameters of the robot's end effector is based on the analysis of images from two orthogonal planes, and it can be fully automated.

Following the identification of the kinematic parameters of the robot and its end effector, this Chapter introduces the kinematically augmented non-model-based algorithm for detection of interaction forces, which enables correct interpretation of interaction influences under different movements. The Chapter concludes with discussion of the results and considerations for implementation in standalone and hybrid configuration with model-based approach.

The Fifth Chapter summarizes conclusions derived from the overall discussion and results analysis of the work presented in the thesis.

2. INTERACTION DETECTION

The intention of the research presented in this thesis is to use the existing sensors commonly found in industrial robots and use them with intention of detecting interactions of robots with their surroundings. Industrial robots typically feature sensors which are essential and used almost exclusively for the efficient execution of the direct and indirect kinematics control tasks.

This chapter aims to present and summarize the different approaches in detection of physical interaction of the robot and its surrounding. The approaches will be discussed in terms of their reliability, complexity, sensitivity, and applicability in order to present their respective advantages and disadvantages.

2.1. Approaches to the interaction detection

2.1.1. Model-Based approach

Model-based approaches are by far the most academically represented type of interaction detection algorithms. The primary field of research for these algorithms is collision detection and reaction, but they are applied also in areas such as kinaesthetic robot guidance [15]. There is a very good justification for their popularity in academic papers, and that is their sensitivity. Depending on the quality of the model they rely on, they can detect contacts measured even in fractions of Newtons, without using additional sensors which industrial robots typically do not possess. For that reason, these types of algorithms are popular choice of collaborative robot's manufacturers, since they have the best possibility to obtain or calculate accurate models of the robots they produce. Simultaneously, they allow producers of collaborative robots to omit torque sensors from robot joints, which positively affects both the engineering challenges related to their integration in each joint and the final price.

As their name suggests, model-based algorithms rely on model of the robot and identification of its parameters. A number of successful collision detection algorithms have been proposed so far, and they mostly rely on model of the robot or some form of dynamics model [5], [8], [9], [10], [12], [16], [17], [18], [19], [20]. Some of them include algorithms to identify the model [9], [17], [18], [21], [22] of the robot which they use to predict values of currents or torques. Predicted values are then compared with measured values in order to detect collisions.

Mostly, the model which they require is the dynamics model, including inertia, friction, and rigidity related parameters, which in turn also relies on kinematic parameters of the robot. Whereas the identification of the latter is generally easier to attain, identification of the dynamic parameters is much more complex [23], [24], [25], [26], [27], [28], [29], [30], [31], [32], [33], [34]. While the theoretical and practical procedure for their identification have been known addressed in numerous research papers, there still are limitations to the practical implementation. These limitations are mostly related to the physical and structural possibilities of the robot to perform movements necessary for the successful identification of the required parameters. This is evidenced by the numerous research papers related to the calculation of optimal excitation trajectories [35].

Model-based algorithms can only reach peak performance if they are applied to the robot under same circumstances as when the model was identified. If some relevant piece of equipment or accessory, such as robot tool or wire feeder for welding robot, is changed or was omitted during the identification of dynamic parameters, the entire modelling process may need to be repeated [36]. These issues are addressed to some extent by the possibility of certain robot brands to run a built-in routine for identification of inertial parameters of the load or gripper. However, this feature does not solve the issue of equipment which is not attached to the end-effector, and tools or load which do not allow performing movement required to identify these parameters due to spatial or structural constrictions.

Another drawback of model-based algorithms is that unmodelled interaction forces which occur during intentional physical interaction of robot's end effector during contact tasks may cause improper behaviour. If models of such interactions are not accounted for, they may lead to false collision detections, affecting the production cycle and overall efficiency. Various techniques [10], [12], [18], [21], [37] have been developed to help prevent contact misinterpretation, such as various filtering and analysis of the duration and intensity of detected change, or learning from previous measurements. Some of algorithms are able to identify collided link [8], [17], [37] and react accordingly, in order to minimize the severity of impact or modify robot trajectory or joint configuration. However, the additional analysis and filtering induce additional time needed for interpretation, effectively slowing down the reaction and therefore possibly increasing the risk and severity of the injury.

In practical terms, this drawback is commonly addressed with introduction of force/torque sensor between the robot's flange and the end effector, which does provide useful feedback which can be used to mitigate these issues. However, there are certain limitations to this approach, besides increased implementation complexity, and they are especially related to the nature of the contact task and/or environment where the robot operates. Manipulation of heavy loads is one example where the possibility of integration of these sensors is questionable and complex due to the transfer of weight from the end effector to the robot body. Tasks which expose end effector to vibration or impacts are typically not favourable conditions for implementation of force-torque sensors since it may affect their reliability or longevity. The same can be applied to work in environments with high temperatures, humidity or otherwise unfavourable conditions for which robots have adequate ingress protection (IP) level, but there are no sensors with comparable protection level.

2.1.2. Non-model-based approach

Approaches which do not require a model of the robot to detect its interactions with surroundings belong to a group of non-model-based approaches. However, in some cases it is difficult to constitute a line which determines to which extent can certain approaches be considered as non-model based. That line depends on the perspective of what is considered as a model. In this thesis, non-model-based approaches and algorithms are those which do not rely on parameters and values which have direct corelation with physical parameters of the system. Therefore, to some extent, this definition could include neural network and fuzzy logic-based algorithms which may even be used to predict values of measured signals in robot joints [17], [37].

The obvious advantage of non-model-based approaches is that they do not require complex identification of the model parameters. Consequently, their performance does not depend on the quality of the model which may be difficult, impractical, or even impossible to obtain. Moreover, this type of approaches and algorithms tend to be easier to implement in production environment since they are less likely to require specialist knowledge related to model identification and related experiment execution.

Arguably the most important advantage of non-model-based approaches is that they implicitly include the entire dynamic behaviour of the system they were developed and implemented for. This means that all interactions, process-specific phenomena, and behaviour of the system are accounted for and can potentially be used for the interaction detection, interpretation, and reaction. From the viewpoint of detection of interaction forces, this is a very important aspect, and the one which has a lot of potential if correctly implemented.

The drawbacks of non-model-based approaches mostly relate to their lower sensitivity and versatility. Their lower sensitivity for detection of interactions, compared to the model-based algorithms, is mostly caused by the uncertainties of synchronizing the measurements from the robot with their corresponding expected values. Additionally, their versatility is limited by a predefined set of measurements they rely upon to detect and interpret interactions. If an action of the robot or the type of interaction with surroundings is out of scope of the set of measurements based on which

the detection and interpretation is performed, the algorithms may behave unexpectedly or incorrectly.

The drawbacks of non-model-based approaches made them less attractive for researchers looking for universal solution for interaction detection. The strive to develop a comprehensive model-based solution, makes it easy to overlook the great potential that other approaches have, especially given the properties of the industrial environment in which they are arguably needed the most. Namely, the predominant types of applications for industrial robots include repetitive and pre-defined tasks with a limited set of variations. That sort of environment favours the reliability, simplicity and applicability of the non-model-based approaches while minimizing the impact and importance of their drawbacks.

The work presented in this thesis aims to explore and harness the potential of non-model-based approaches for implicit inclusion of dynamic behaviour of the system by developing reliable and applicable algorithms for interaction detection. Equally importantly, the intention is to develop solutions for drawbacks of the non-model-based approaches and present them as viable or even preferable alternatives to the model-based algorithms.

2.1.3. Hybrid approach

In some respects, it is possible to notice that the two general approaches have complementary advantages in terms of complexity and implementation. A hybrid approach, combining advantages of both the model- and non-model-based type of algorithms is also examined and presented in this thesis. The main idea behind the proposed hybrid approach is to augment the non-model-based approach with coefficients corresponding to identified kinematic parameters of the robot. The kinematically augmented algorithm's promising properties have the potential not only to expand the field of application of the non-model-based approaches, but also to solve some of the drawbacks of the model-based algorithms related to the contact interaction tasks. Another dimension of hybrid approach which will be theoretically examined is to use model and non-model based algorithms in a cascading fashion. In this configuration, model-based algorithms would be used to cover nominal behaviour of the robot and its task, while the non-model-based algorithms would be used to complement them by incorporating the unmodelled or unforeseen interaction dynamics.

2.2. Analysis of available signals

The first step in understanding the possibilities and considering the design of the interaction detection algorithm is the analysis of the properties of the available signals. This section aims to provide insights into characteristics of signals that are commonly available in industrial robotics.

Industrial robot controllers with closed control architecture rely on two main types of signals for efficient control of robot manipulators. These types of signals are necessary for achieving any meaningful robot behaviour and are consequently present in all types, configurations, and generations of industrial robots with rotational joints.

The first readily available type of signal is the information about each of the joint angles, referred to as joint coordinates, and typically represented in degrees. Industrial robots almost exclusively receive this type of information from incremental encoders, although there are also other realizations employing absolute encoders and resolvers. This type of signals is used in control algorithms for both direct and indirect kinematics, along with angular velocity and acceleration, which are obtained as derivates and double derivates respectively.

The second type of signals which are used in control loops is the measurement of joint currents. The range of their values depends on the rated values for the joint motors, and depending on the robot brand, they are either available in Ampers or percentage of the joint current rated value. Depending on the manufacturer, both representations may include the information on polarity or be represented in absolute values. Some manufacturers also offer another signal, and that is the torque values which are estimated based on the current measurements. Similar to joint currents, these estimates may be represented in absolute values of Newton-meters, or as percentage of the permissible joint torque values, but are almost always signed, i.e., have positive and negative values.

Having in mind that the predominant architecture of industrial robots employs rotary joints, the effect of external forces of interaction on robot links translates into torques affecting the joint motors. At the joint actuator level, these torques manifest themselves in form of changes in joint current, indicating the presence of external force. The nature of the environment and tasks for which industrial robots are intended for has led to the design of control algorithms which efficiently and promptly eliminate the effect of the disturbances on the commanded joint position values. Consequently, the changes in joint positions caused by the external forces are rarely allow noticeable. For this reason, the primary source of interaction-induced information in industrial robotics are the measurements of joint currents, and the torque estimates based on them.

Following sections will present the analysis of available and subject-relevant signals, with focus on joint currents and torques obtained from the industrial robot with closed control architecture. The robot used to illustrate signal properties is Denso VP-6242, a 6-axis industrial robot. Measurements from this robot are provided on a scale -100% to 100% for joint torques, 0% to 100% for joint currents and in degrees for the joint angles.

2.2.1. Repeatability analysis

Signal repeatability in the time domain is one of the most important aspects and preconditions for designing efficient non-model-based interaction detection algorithms. This signal property can be observed from the perspective of the robot itself i.e., its actuators and the equipment permanently attached to it, as well as from the nature of the desired contact task. The following results are related to the properties analysis of the former, while the latter will be addressed in the following sections on a case study-basis.

With the aim of discussing this important property of the signal, measurements of joint values were recorded while the robot was performing the same repetitive task in cycles., and results were analysed. Thirty consecutive measurements from the same movement are shown on Figure 1, and they show that the overall shape of the signal is generally repeatable, which is favourable from the standpoint of the non-model-based algorithm development and implementation. However,

although the beginnings of the signal were matched to start at the same time instant, it is evident that they start to deviate from each other after only a few samples, and it is important to find the cause for this observation.

Robot brands typically possess some form of parallel processing capability. However, these apparent parallel processes are often performed on single core processors, meaning that they are not strictly run in parallel. Instead, they share resources of the single processor by switching rapidly from execution of one process to another. This fact causes slight deviation of time instants in which identical command will be executed within a repetitive cycle. Therefore, it is very difficult to guarantee that measurements obtained in one of parallel processes will be performed with strictly accurate periods. That in turn means, that sampling of measurements will not be constant, but some periods will slightly vary from the others.

This also includes the motion commands, which consequently leads to the situation in which measurements from consecutive measurements from the identical movement do not match in time domain. Additionally, it is clear that one cycle of robot movement doesn't necessarily contain a whole number of samples. When all aforementioned is put together, the result is that movement of the robot will not be sampled in same time instants. The consequence is that measurements from two identical movement cycles of the robot cannot be compared directly in an effective way. On the left graph of Figure 1, these differences are shown as set of 30 consecutive recordings of the torque measurements in a robot axis for same movement sequence, all varying in duration.

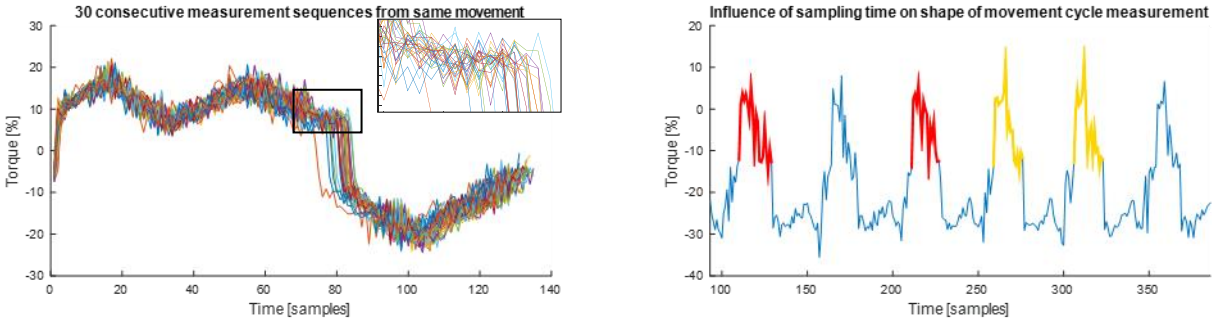

Figure 1. (left) Measurements from 30 consecutive executions of the same movement [38] show time domain related differences in recorded signal caused by the differences in sampling instants. Signals have slightly different lengths, and they are shifted in time for up to 13 samples or just under 10% of the total length of the signal. (right) Six consecutive movement cycles and differences between them which are caused by sampling. Signal peaks shown in sections of signal marked in yellow are not visible on sections marked in red. Peaks marked in yellow and red show biggest effects of the issues related to numerical position derivation.

Another observation related to the nature of the measurement signal of current/torque is the irregular occurrence of peaks, as shown on the right section of Figure 1. The occasional appearance of peaks in measurement is related to the fact that currents/torques are dependent on speed and acceleration in each joint [13],[15]. As mentioned earlier, due to the fact that industrial robots possess only position measurements from encoders, speed and acceleration are calculated numerically as derivatives and double derivatives of position measurements. Depending on the actual dynamics of the position change in relation to the sampling rate, these numerical derivations may result in values which do not correspond accurately to the actual dynamics of the robot [39]. Although these numerical values may differ from real values, they are absolutely adequate for the proper functioning of the robot due to robot's inherent actuators and structural inertia which filter out the high frequencies of the control signal. However, from the standpoint of observation of signal deviations, these peaks may be mistaken for unexpected external forces. The peaks must not be filtered out because they may contain important indications of a collision or other real external force/torque which are of importance for the collision detection algorithms and worker safety.

2.2.2. Statistical signal analysis

Detection of interaction of industrial robots with their surrounding requires setting certain thresholds and limits which are used to determine whether certain interaction has occurred or not. The thresholds for such detections can be determined in various ways, but all of them require understanding of the factors which influence their values. For better understanding of these influences, the following analysis is conducted on first three joints of the 6-axis industrial robot to demonstrate effects which position, speed and acceleration have on signal properties. These joints were selected specifically because they contribute the most to the positioning of the robot end-effector, and therefore best illustrate these influences.

In the first test phase, for each joint, both currents and estimated torques were recorded along with the joint positions which are used primarily to better depict different stated during measurements. Each joint was moved individually, and the same movement was repeated with 5%, 10%, 20% and 40% of the maximum speed of the robot, to which corresponds 0.25%, 1%, 4% and 16% of the maximum acceleration.

The measurements from the 1st axis shown on Figure 2 show that, both for the joint torque and the current, the influence of the joint position does not have any noticeable effect in the stationary state, when the angle does not change. This observation in due to the fact that the 1st axis of the robot is vertical and is therefore unaffected by the weight. The influence of speed and acceleration is evident in each of the transitions to the new position, and it is quite repeatable in shape for the same reason. In this example, the current and the torque have similar shapes, and their real physical correlation is evident. The exception can be noticed at the beginnings of each sequence, since the torque has negative values, whereas the current is shown in absolute values. The values of the variance in the stationary state are $\sigma^2 = 0.1283$ on average for the current and $\sigma^2 = 0{,}1452$ for the torque for the 4 regions without change in angle.

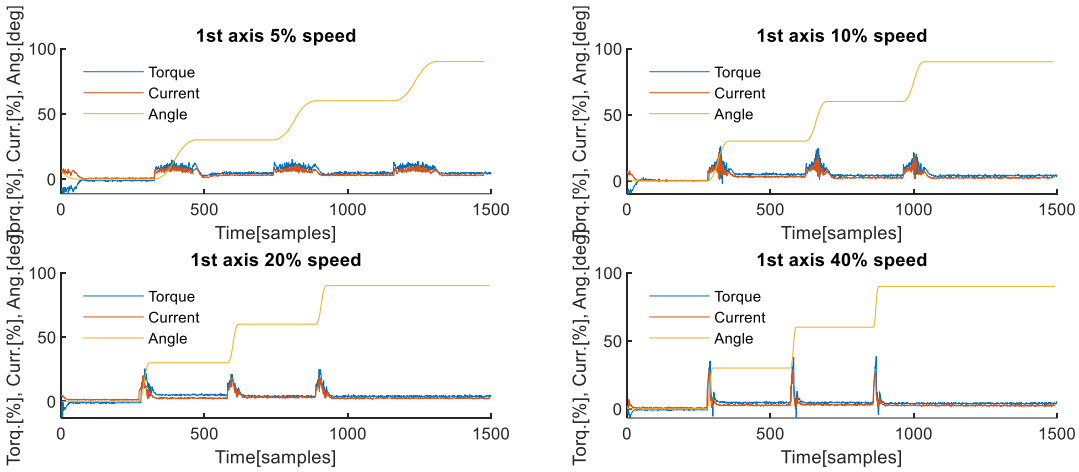

Figure 2. Influence of the joint position and speed on measured values from the 1st axis. The graphs illustrate that the stationary currents and torques for the 1st axis are mostly unaffected by different positions due to the invariance to the influence of gravity and unchanged weight distribution. The effects of speed and acceleration differ only in the periods with movement.

The 2nd axis of the robot presents same type of measurements as shown for the 1st axis, but performed in both directions, for increasing and decreasing joint values. Since the 2nd axis is horizontal, the measurements show differences between the upward and downward movements, caused by the different weight distribution and its effect on joint torques and currents. Results shown on Figure 3 show that the different positions do not affect the variance of the current and torque signals in stationary states, which are $\sigma^2 = 0.0631$ and $\sigma^2 = 0.0029$ respectively. Transient periods show that profiles of currents and torques during the movement differ from each other in

shape, which is caused by the different weight distribution of the succeeding links and joint. When observed for the same portion of the signal, it is evident that shapes of torques and currents are scaled mirror images of each other, with exception to the sections where differences exist due to availability of absolute values of the currents.

The same conclusion as with the upper row of Figure 3 can be made with the lower row of the same figure, on which the values recorded during upwards motion were recorded. The variance in the stationary sections is almost identical to the previous analysis, but the variance in the transient periods is smaller.

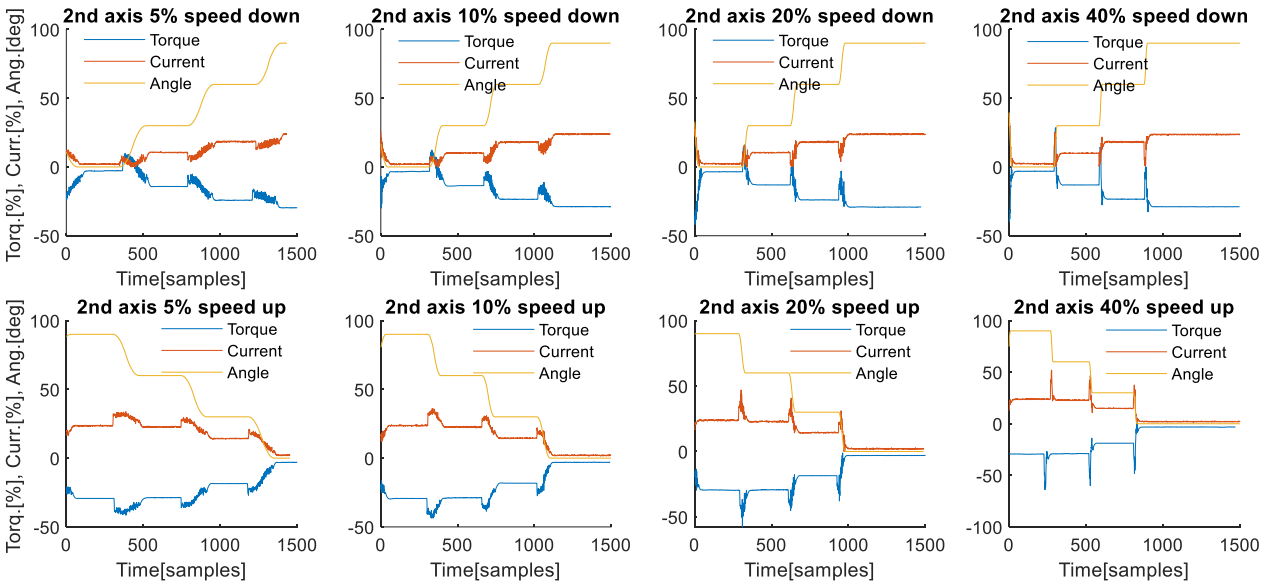

Figure 3. Influence of the joint position and speed on measured values from the 2nd axis. (upper row) The graphs during downward movements illustrate that the stationary currents and torques for the 2nd axis increase in mean values due to the influence of gravity and changes in weight distribution. The effects of speed and acceleration also change with changes in robot joint configuration. Torques resemble scaled mirror images of currents in regions where the current does not change sign. (lower row) The graphs during upward movements illustrate that the stationary currents and torques for the 2nd axis decrease in mean values due to the reduced influence of gravity and changes in weight distribution. The effects of speed and acceleration also change shape with changes in robot joint configuration. Similar to the downward motion, the torques resemble scaled mirror images of currents in regions where the current does not change sign.

The conclusions made for the analysis of the 2nd axis are also valid for the analysis of downward and upward motion of the 3rd axis, shown on Figure 4. The difference in tested angles is caused by the physical constructions of the robot, which limit the motion of the 3rd axis, not allowing angles smaller than 19 degrees. The variance of measurements in stationary periods is almost identical for each joint angle and is equal to $\sigma^2 = 0.1125$ for joint currents and $\sigma^2 = 0.2131*10^{-3}$ for the joint torques.

For the next phase of testing, each of the axis was moved individually in both directions, with increasing speed. The first movement was made with 5% of maximum speed, while the following movements were made with 10%, 20% and 40% respectively. The Figure 5 shows recorded joint currents and estimated torques, as well as angular joint speeds which were calculated offline based on the recorded joint angles.

Results from the measurement from the 1st axis show that the speed and acceleration do have influence on the variance of the signal, as shown in Table 1 and Table 2, and that higher values of calculated speed have higher variance as well. Closer observation of the calculated angular speed reveals occasional peaks related to the numerical derivatives and irregular sampling periods, which is even more evident from the analysis of the lower left graph showing angular speed recorded with the current measurements. Another observation which can be made for all

graphs is that measurements of torques shown in the upper row, and currents, shown in the lower row, have up to 10% different numbers of samples, although they belong to the same movement sequence, providing further proof about the differences in sampling periods.

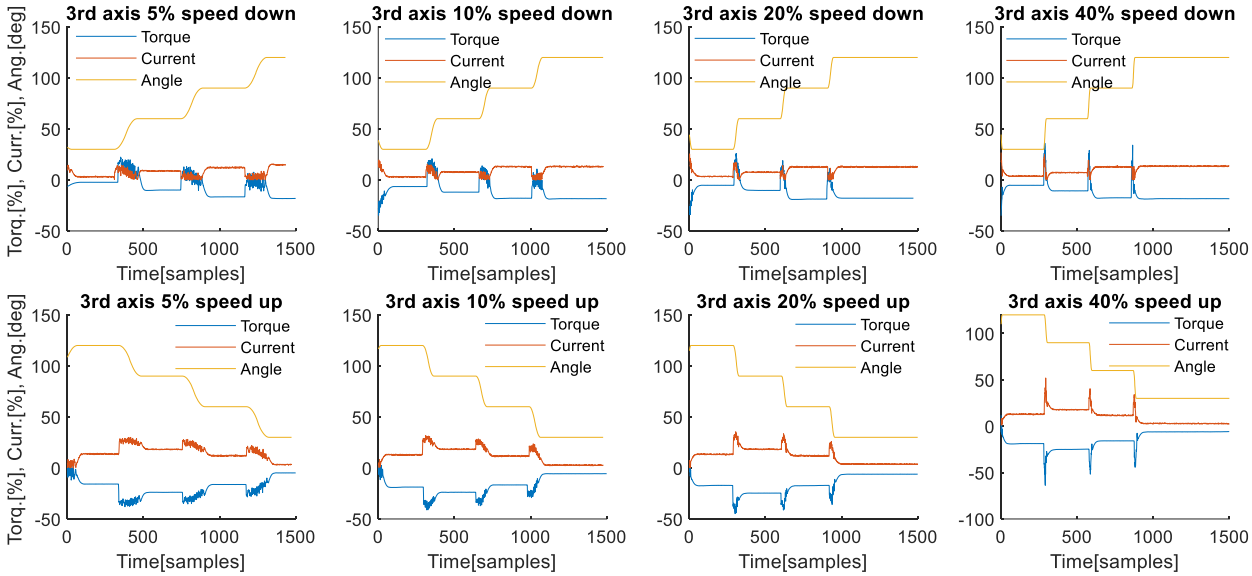

Figure 4. Influence of the joint position and speed on measured values from the 3rd axis. (upper row) The graphs during downward movements illustrate that the stationary currents and torques for the 3rd axis change in mean values due changes in weight distribution. The effects of speed and acceleration also change with changes in robot joint configuration. Torques resemble scaled mirror images of currents in regions where the current does not change sign. (lower row) The graphs during upward movements illustrate that the stationary currents and torques for the 3rd axis reach highest mean values at the angle of 90°, since the centre of the mass is at its furthest position from the observed joint. The effects of speed and acceleration also change shape with changes in robot joint configuration. Similar to the downward motion, the torques resemble scaled mirror images of currents in regions where the current does not change sign.

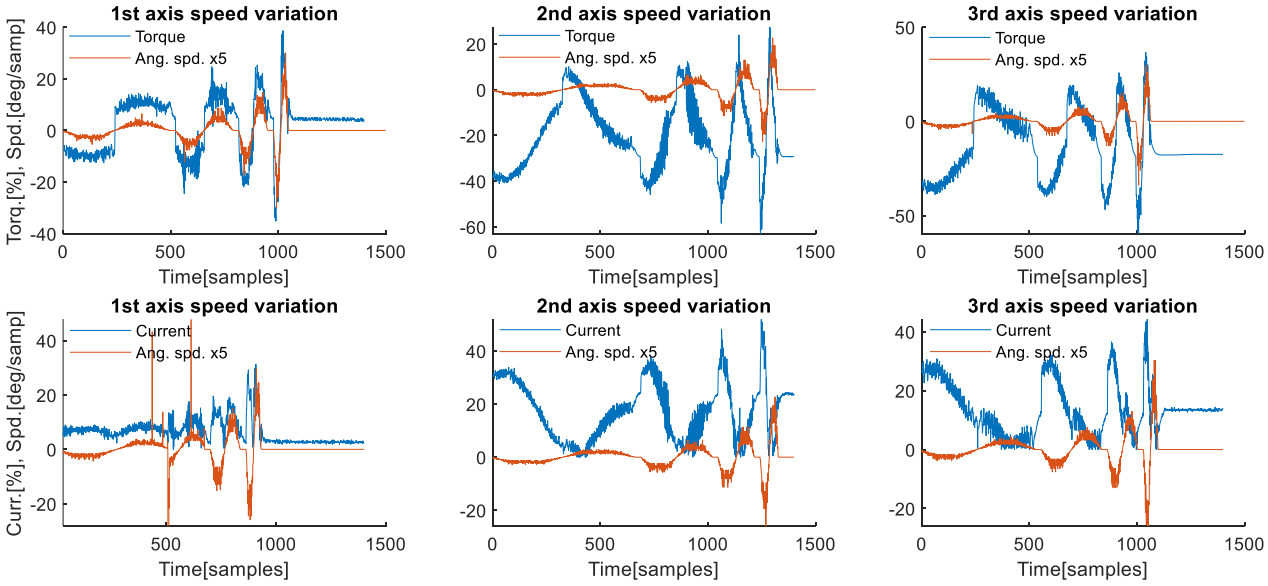

Figure 5. Influence of the joint speed on measured joint values. (left column) Effect of the speed on the 1st axis measurements shows that the speed has non-linear influence on the variance of the measured signal at the specific sampling frequency. Since the axis is unaffected by the gravity, profile of torques shown in the upper part is symmetrical. (middle column) Influence of the changes of speed is non-symmetrical due to the different 2nd joint motor effort in upwards and downwards motion. Upward motion results marginally in higher signal variance. (right column)) Influence of the changes of speed on the 3rd axis is non-symmetrical due to the different joint motor effort in upwards and downwards motion, which is better observed on the torque graph. (upper row) Profiles of the torques are more

intuitive to the analysis since they have both positive and negative values. (lower row) For the same movement, duration of the sequences with current measurement is different from those of the sequences with torque measurement, shown in the upper rows.

Table 1 Influence of speed on the standard deviation σ of joint currents and torques

	1st axis		2nd axis		3rd axis	
Speed	Current σ up/down	Torque σ up/down	Current σ up/down	Torque σ up/down	Current σ up/down	Torque σ up/down
5%	1.1125/0.7512	1.3106/1.1674	0.2477/0.2486	0.1182/0.0538	0.2591/0.2632	0.1743/0.1904
10%	1.8436/3.0848	2.5471/2.0940	0.2233/0.2790	0.2204/0.3109	0.2460/0.2143	0.1927/0.2213
20%	2.3864/1.6495	2.4470/1.9838	0.2359/0.2381	0.1388/0.1302	0.2381/0.2691	0.2703/0.2148
40%	1.6772/4.3021	4.5836/3.4267	0.1875/0.2173	0.0120/0.0014	0.2429/0.2942	0.5770/0.7226

A sequence featuring movement at different speeds and to different positions, will have sections with different joint measurement dynamics. Although variance across the sequence will differ, each section generally conforms to Gaussian/Normal distribution, and therefore standard deviation σ can be used to determine the limits within which all measurements should occur. However, in cases where a sequence feature underrepresented and drastic changes in the dynamics, induced, for example, by the changes of the motion speed as in Figure 6, the variations are not always within the range defined by the standard deviation, as shown in red. Consequently, the conformity to 3σ interval, is best applicable to the regions with similar variance, as show in yellow on Figure 6. Therefore, in such signals the limits may best be determined in a segmented fashion, with each segment corresponding to the region with similar variance, as shown on Figure 6. Alternatively, the variance may be observed on a sliding window basis, width of which depends on the performed task, as shown in purple on the same figure. The latter is by far the most appropriate for the applications in which deviations need to be detected, as such limits offer better sensitivity. However, it is important to note that the detection of interactions often involves numerous compromises related to implementation and complexity, including those related to detection thresholds.

Table 2 Influence of acceleration on the standard deviation σ of joint currents and torques

	1st axis		2nd axis		3rd axis	
Acc.	Current σ up/down	Torque σ up/down	Current σ up/down	Torque σ up/down	Current σ up/down	Torque σ up/down
0.25%	0.6823/0.6649	0.7404/1.1751	1.1783/1.4022	0.7325/1.3214	17522/2.1761	2.4862/3.4641
1%	0.8164/3.0848	0.7196/0.5693	1.4194/1.3965	1.5532/1.4382	2.1201/2.1691	3.2297/2.9094
4%	1.7844/1.5855	1.9492/0.9976	3.3351/2.8852	3.2123/2.6564	2.7813/2.6931	4.0892/4.0996
16%	1.6762/2.6651	3.5512/4.2231	4.1914/2.8623	5.7054/4.6271	6.0691/5.2112	6.1269/4.7553

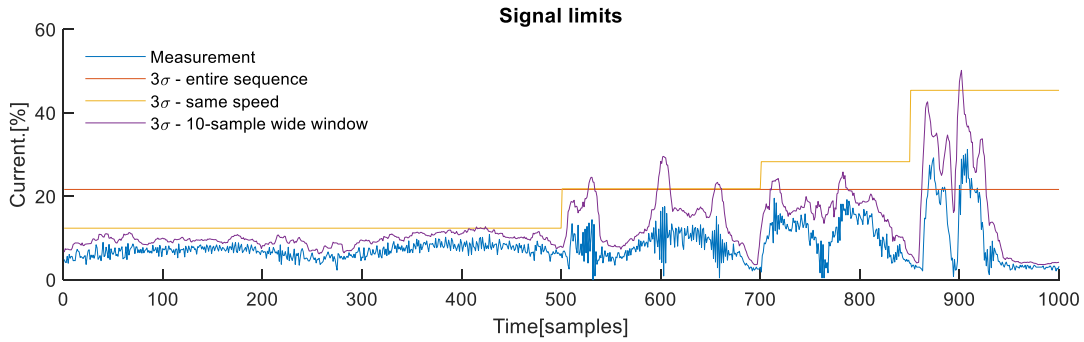

Figure 6. Signal limits determined on three standard deviations from the mean value. The figure illustrates the effect of the observed interval of the sequence to the conformity to Gaussian distribution. The best results are achieved when sections with similar speed are observed, or when sliding variance window principle is applied.

The previous analysis has shown the effect of the position, speed, and acceleration on the measurement variance. Industrial robots are relatively complex machines with intricate correlation and cross-influence of their integral parts. The following analysis aim to illustrate some of these cross-influences using an intuitive and simple example.

For the two joints whose axes are not perpendicular to each other, movement of one joint affects the joint currents on the other joint. Such effect can be best observed in joints with parallel axes, such as 2nd and 3rd joint, measurements from which are shown on the Figure 7. The left portion shows the effect movement of the 2nd axis has on currents and torques of the 3rd axis. The right portion of the figure shows the opposite situation, in which the 2nd axis is stationary, but still affected by the movements of the 3rd axis.

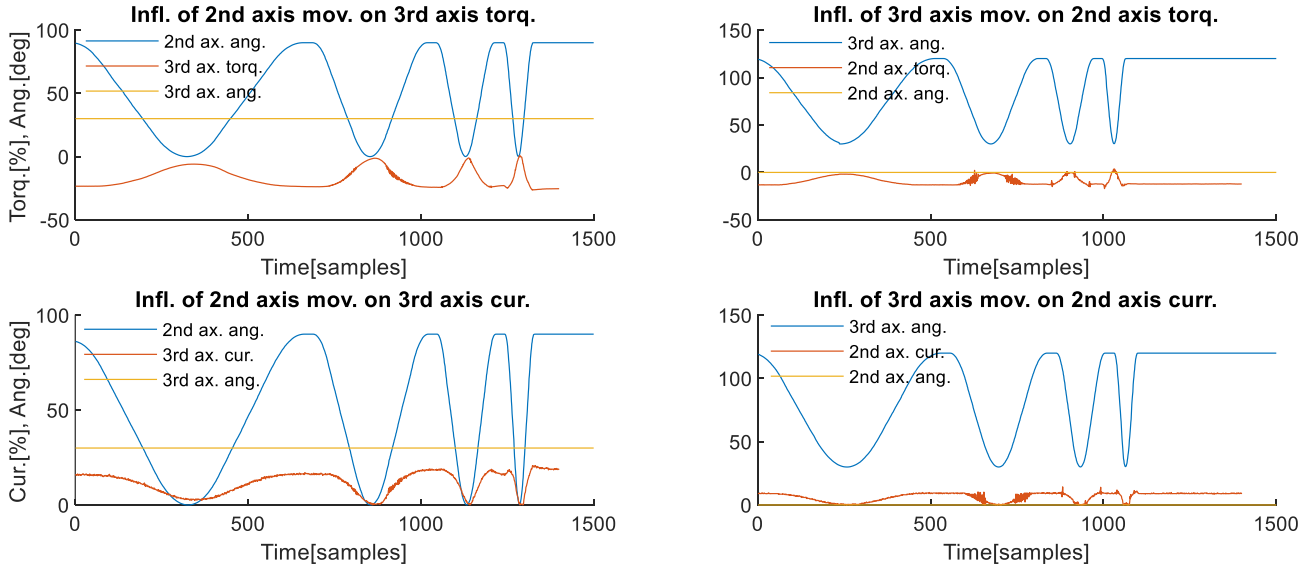

Figure 7. Cross-influence of the torque of stationary joints observed on an example of joints with parallel axis. (left column) Effects of movement of the 2nd axis reflected on torques and currents of the 3rd axis. (right column) Motion of the 3rd axis influencing torques and currents of the 2nd axis while it was not moving.

Results from Figure 7 illustrate that the motion of one axis has manifested itself onto a different stationary axis in form of disturbances in mean value of currents/torques, as well as in some occasional influences on the variance. These influences are a consequence of the shifts in distribution of the mass and the inertia experienced by the observed stationary axis. However, although the disturbances differ in dynamics, they have similar peak values and similar shape, though stretched and compressed in time domain. This indicates that at speeds up to 40%, movement acceleration impacts on the torque are much less prominent than the influence of gravitation acceleration, and consequently the weight distribution. This conclusion will be important for the analysis in the 4th Chapter, where spatial relation of joints is considered.

From the aforementioned observations, it is possible to conclude that industrial robots with closed control architecture pose a unique set of challenges when it comes to detecting deviations from their expected joint current values. When these challenges are viewed from the perspective of interaction detection, it should additionally be noted that all signal processing and decision making must be made in very short amount of time in order to make timely reactions of the robot and prevent injury or damage.

3. DETECTION IN REPETITIVE TASKS

The traditional philosophy of automation still often divides automated and robotized processes from humans. There are numerous reasons why the division has been considered desirable and even necessary, but the most important reason is safety of the human workers. Although industrial robots have come long way since they were first introduced in production, for a long time they were developed to work in human-free environments, and hence the safe interaction was not a major concern.

Prerequisites for safe physical human-robot interaction are described in three nested layers which robot must guarantee – safety, coexistence, and collaboration [8], [9]. Safety, the innermost layer, relies on various internal and external sensory, mechanical construction, control algorithms and speed of the robot to reduce the risk of injury to the worker and of damage to the robot. Consequently, lot of research [3], [4], [5], [16], [40], [41], [42], [43] has gone into standardization and setting safety limits for human-robot interaction. Collision detection is one of the key aspects of the safety layer, and it is one of the few features of safety possible to effect on an industrial robot without any external sensing and closed control architecture. Efficient collision detection also enables sharing of the workspace with humans, and opens possibility for coexistence, since humans and robot can perform separate tasks in the same environment. Collaboration requires physical interaction between human and robot in such way that the robot can sense the intentions of human, and react accordingly, as well as to exchange information with the worker in an intuitive and predictive manner [44], [45]. As such, collaboration is the next from coexistence, and is therefore also dependent from reliable collision detection.

One of the basic ideas of the algorithms proposed in this Chapter is to use the fact that most present robot tasks in industry are highly repetitive. This fact enables recording of one typical movement cycle of the robot and using recorded values as reference current/torque sequence. Comparing reference sequence with measurements obtained from the robot during its operation enables detecting deviations from reference values. Collision detection algorithms proposed in this Chapter and in [13], [38] rely on these principles in order to detect collisions and react accordingly. Although these algorithms are not as versatile as model-based algorithms in terms of modification of robot movement, they offer advantages in reduced complexity and reaction time. Additionally, since reference sequences are recorded from the real operation cycle of the robot, they inherently include changes to the current/torque profile induced by picking and placing, assembling, as well as other intentional contacts.

3.1. Implementation on the robot controller

The first algorithm for interaction detection to be presented and analysed in this thesis is intended for implementation on the industrial robot's controller with closed architecture and limited capacities for connection to external devices. As mentioned before, the closed control architecture of controllers is typical for most robot brands, and it often allows very basic structures and types of commands. Compared to [12], [16] [19], [21], [10] and in particular similar method used in [18], this section presents an algorithm which can be fully implemented inside robot controllers with closed control architecture, without torque or any other external sensors or devices, as a contribution to safer human-robot coexistence.

The idea behind implementation of the algorithm on the controller itself is that it could positively influence the reaction time to the detected interaction. Gathering, arranging, and sending information to external devices and waiting for processing and feedback regarding the detected interaction may consume valuable reaction time, especially with older generations of robots, whose processing power is not substantial. Instead, the intention is to use some of the commonly used communication protocols to perform data acquisition, analysis, and setup in conjunction with a PC, and then apply a simple detection algorithm on the robot. The algorithm relies on readily available joint current or torque measurements, and the ability to stop the robot motion with a high priority level command or hardwired signal. The relative simplicity of the algorithm ensures its fast execution and minimal processing effort on the controller's side, but it is also very convenient as an introduction to the idea of non-model-based approaches.

3.1.1. Principal Design

The design and background idea of the presented algorithm imply that the field of its application are robot tasks which involve movement which repeats in cycles in identical ways and under same circumstances. Consequently, the measurements of currents or torques on a joint level are also expected to repeat in values cyclically. The main idea of the algorithm is to record these values during execution of one representative cycle are then use them as reference values for all following cycles. All deviations of the measured values from their corresponding reference values indicate that there is an unexpected interaction with an external force. The algorithm itself cannot determine the nature of the unexpected interaction, and therefore all deviations from the reference values are considered to be collisions. For that reason, the approach presented here effectively describes the working principle of the collision detection algorithm. It goes through steps in execution of the algorithm and explains how the collision is identified on a joint level, with the following description being valid for every joint.

The algorithm contains two procedures, out of which only one is executed at any time, and there are two main phases required for the successful implementation of the interaction detection algorithm.

The first phase is related to the actions needed for the proper setup of the algorithm, setting the reference values and tolerances. In the setup phase, the first procedure is executed only once, and its sole purpose is to record joint torques while robot is performing its main, repetitive task. During execution of the first procedure, torques from several periods of repetitive movement are stored in internal memory of the robot. The recorded values reflect the effects which the internal and external forces have on joint during a typical cycle of the repetitive task execution. For this reason, they are used as reference values to which measurements from all subsequent cycles of repetitive task execution will be compared against in order to detect potential unexpected interaction.

The second procedure has dual purpose, and it is used both in setup and operation phase. In the setup phase, the second procedure is executed after acquisition of the reference values and used to determine the number of samples in one period of execution, which is later used for resampling purposes. The reason for using the second, rather than the first procedure to determine the number

of samples is that this procedure will also be used in operation phase, unlike the first procedure. The determined number of samples, which depends on the complexity of the executed procedures, will therefore represent number of samples which will be encountered in operation phase.

In order to perform measurements reliably, both procedures rely on triggering signal from the main task of the robot, which is used to ensure synchronization of measurements with the periodic movement of the robot.

Once the measurements are acquired/recorded, they are transferred to PC for processing, where they are analysed in terms of dynamics and signal limits. Processing includes generating reference limits of the torque for one period of task execution, as well as limits related to the dynamics of the signal. After reference limits are determined, they are transferred to the robot's controller to be used for collision detection, from which point onwards, the connection with PC is not required.

The second phase of the algorithm is related to the actual operation of the robot with interaction detection functionality. In this phase, only the second procedure is used and its main assignment is to compare online measurement values of the torque with reference limits in order to detect collision/interaction. Collision detection is a result of at least one of two conditions being met at any joint. First condition, or first decision rule, is that measured torque is within expected limits set by the PC, and it is more sensitive to detection in periods when acceleration of the robot is not changing rapidly. Second condition, or rule, is related to the dynamics of the torque signal, and it demands that difference between two consecutive samples is smaller than a certain value. The second condition is more restrictive in periods with higher torque dynamics, i.e., when acceleration changes sign or values rapidly.

Since it was shown in [12], [16], that collision on a segment influence is most prominent on joints that preceded it, the implemented algorithm can also detect where the impact has most likely occurred. Although this information can be useful in many situations, in current implementation it is used only to verify the validity of the collision detection in testing phase, and as such is not subject of any further analysis or discussion.

Algorithm and both of its procedures are designed to run as processes parallel to the main robot task. When robot is performing its main routine in the operation phase, the second procedure is running in background, and its only task is to stop the movement of the robot and signal to the main routine when collision is detected, as shown on Figure 8.

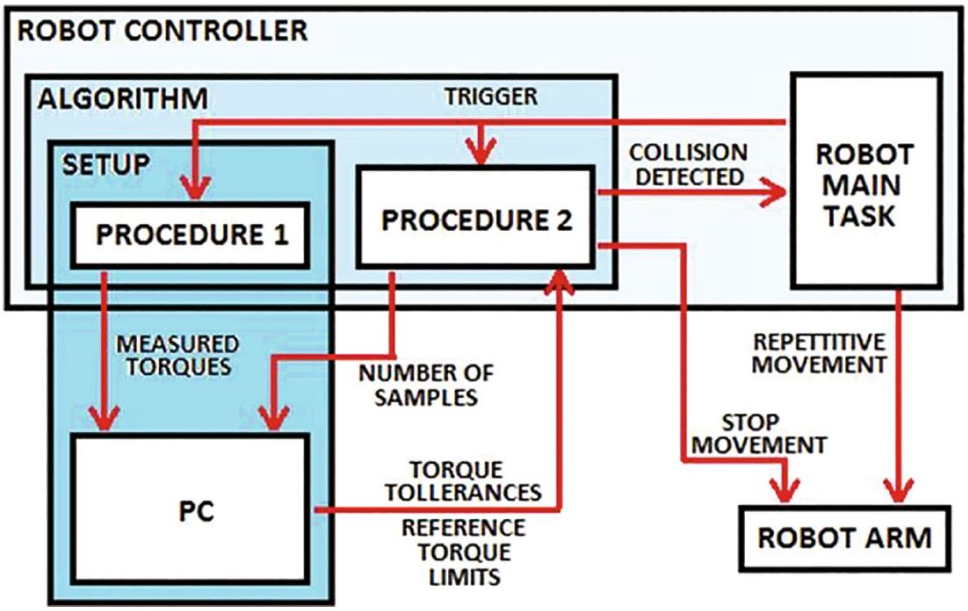

Figure 8. Structure of collision detection algorithm [13]. The algorithm requires PC only during setup, and it can stop the movement of the robotic arm independently of the main robot task

3.1.2. Setting reference limits and thresholds

Analysis of the signal was conducted on a set of samples from multiple measurements. Values of tolerances and signal dynamics mentioned in following text are particular values for the performed sequence. However, corresponding values for decision rules can be generated for any given sequence.

The left graph of Figure 9 shows thirty execution sequences of the task while the right graph shows absolute differences in neighbouring samples. By analysing sequences shown in Figure 9, it is possible to make two observations, both of which are results of varying number of samples in one period of execution. First, that the torque signal might appear to be stretched, compressed, or shifted in time compared to other recorded sequences, which is the primary cause for differences between any two sequences at any sample point. However, within one period of cycle execution, torque signal preserves its general shape. The second observation is that the moment when acceleration/torque changes its sign is not constant.

Due to the design of the robot's controller discussed previously, it is impossible to affect the time when torque is sampled and improve the situation, so the focus is on finding adequate compensation.

With regards to analysis of graphs in Figure 9, it is possible to differentiate two different regions based on dynamics of the signal. First region is where acceleration of any joint is relatively slowly changing, and the second region is where it rapidly changes its intensity and sign. First region can be identified on right graph of Figure 9 as period with differences in neighbouring samples of less than 12%, and the second is marked by the regions with changes of more than 12%. The two mentioned regions require two different strategies for identifying collision.

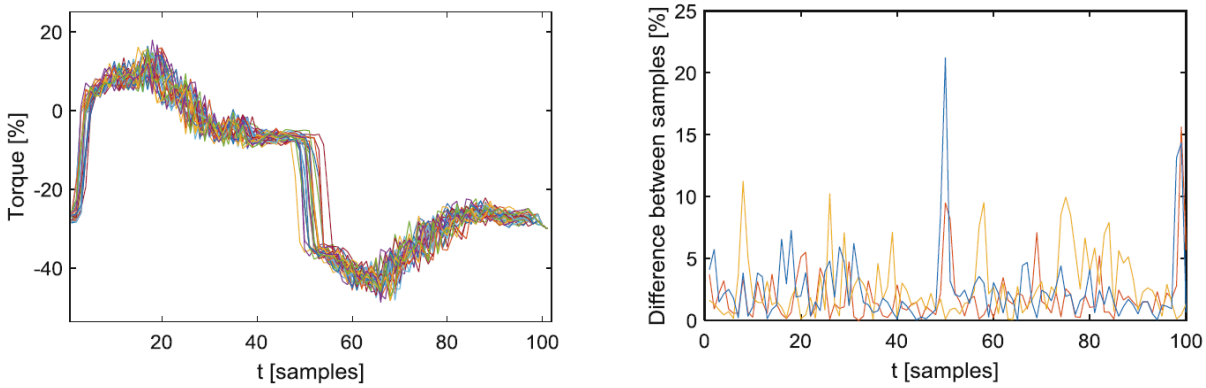

Figure 9. Left graph shows sample of 30 consecutive periods of the torque signal of the third axis [13]. Differences in successive samples over one period of the signal, shown on right graph, differentiate regions with higher and lower dynamics.

From the first observation, it is possible to conclude that the envelopes of measurements shown in left graph of Figure 9, can be approximated relatively well, and generated from just one signal shifted in time backwards and forwards. The maximum number of samples for shifting at any direction can be obtained from second observation, i.e., from the region with rapid change in torque. From the left graph of the Figure 9, it is possible to measure that difference of first and last falling edge of the signal in this region is 8 samples. If the leftmost signal is shifted 8 times for amounts from 1 to 8, the upper envelope at any point can be approximated as maximum of at any point of all the shifted signals. Similar thing can be done for lower envelope and minimum at any point. The two envelopes are shown on left graph of Figure 10.

Statistical analysis has shown that the variations in amplitude of any sequence outside this envelope do not exceed 3%. Since it was conceived that the execution of algorithm during setup is performed only once, it could have happened that the only signal obtained was signal similar to the rightmost sequence on Figure 9 on the left. In that case, the envelope would have been created by shifting the signal 8 samples to the left. Since it is not possible to be sure from which extreme

sample sequence may be obtained, the way to be sure envelope will contain all the signals is to shift the sample sequence both left and right. The statistical analysis of testing for the region critical for this occurrence on more than hundred recorded sequences has shown that the time shifts conform to the 3σ interval from the mean value and the amount of shifting was therefore set to rounded amount of 10 samples.

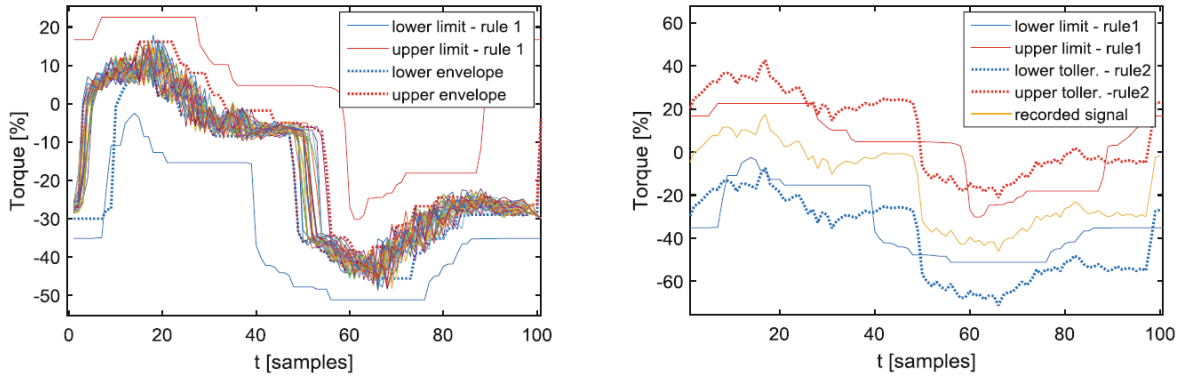

Figure 10. The left graph shows limits for the signal which allow for different number of samples per period, as well as approximation of envelope for all shown signals [13]. Right graph shows limits to the signal imposed by two detection rules, with second rule narrowing transition region.

With respect to analysis of deviation of signals from the previously described envelope approximation, the first rule, or condition, for detecting condition can be derived. Upper limit for the signals can be set as values of upper envelope increased for 5%. Similarly, lower limit for signals can be calculated as values of lower envelope reduced for 5%. Signal limits are shown on Figure 10 as rule 1.

The second rule, or condition, for detection is intended to identify collisions in areas where torque has higher dynamics, i.e., were acceleration changes rapidly. Widening the regions where rapid changes occur means that some collisions may remain undetected, as it will be shown in evaluation section. From the right graph of Figure 9, it is possible to notice that in areas of interest, difference between two neighbouring, or succeeding samples is never greater than 22%, which was also confirmed by testing on a larger sample of sequences based on which the 3σ interval was calculated and rounded to the first higher integer value of 25%. This observation was used to form a second rule, or condition for detecting collisions. If absolute difference between neighbouring samples is greater than 25%, the second rule will indicate collision.

3.1.3. Implementation of detection rules

Processing of obtained torque values is one of the most important tasks in the entire algorithm since its results are used as reference tolerances and limits based on which the collisions are detected. There are various aspects of the processing, and each one of them will be explained separately.
One of the main issues is adaptation to different sampling times which correspond to each of two procedures of the algorithm. Although both procedures have similar structures, execution of one cycle in each one of them is different, but relatively procedure-consistent. Therefore, if each procedure executes command for torque measurement once per its cycle, it is inevitable that these torque values will be sampled with different periods. Since robot's controller doesn't have the ability to perform tasks with strict time execution, i.e., to perform tasks with absolutely exact periods, these periods are considered sampling times in this thesis. In order to compare corresponding samples to each other, it is necessary that sampling times of reference and execution sequences are identical, or very close. To do so, it is needed to measure period of each sequence.
However, due to the fact that period of the robot movement does not contain a whole number of samples, analysis of the signal has indicated that one period of the movement may correspond to

different number of samples. Since number of samples needs to be a whole number, the best possible solution is to pick number of samples closest to the average value. When number of samples per period has been set for each sequence, it is possible to resample reference sequence to match sampling time of execution sequence.
From resampled reference sequence, it is now possible to extract one representative period of signal and use it as a base for further formation of reference limits period. Each period of movement execution contains slightly different number of samples, whose mean value is not necessarily equal number of samples of the resampled reference sequence. Over long time, difference in number of samples may accumulate if the new period of reference signal is started when the previous has ended, and lead to a situation where two sequences are not in phase or may even be in opposite phase. Therefore, in order to prevent potential accumulation of error, it is suggested that main program should trigger the start of each new sequence of reference values. Final step was to create a single period of reference limits by shifting reference torque period in time and adding tolerances to its values, as explained in previous section.

Stopping movement without cutting off the power of the motor is action which can be implemented in various ways on most robots in existence. This enables resuming the motion and execution of the program from the point where it was stopped, once the cause and consequences are no longer present. It is a significant improvement over going through steps needed for going out of the safety procedures for treating post-collision states which may include but are not limited to clearing some signals and warnings, resetting programs, switching on the power to the motors and starting the program from the beginning.

Optionally, the main task can use information about detected collision in order to react to it in designated manner in order to reduce further injury or damage. The robot can react to a number of ways [12], [16] including continuing movement, stopping, moving back, or activating collaborative mode. While algorithm itself stops the movement in event of collision, another reaction has been implemented on the robot, which is to move away from direction of detected impact. This is possible thanks to the use of torque measurement, which carries the information about the sign of deviation. This information can be effectively used to determine from which direction has disturbance occurred, and it is available for all joints. When collision is detected, the algorithm stops the movement, and measures differences between set reference limits and last measured values of torque in every joint of robot. These differences are than scaled and used to indicate how much the robot should move at every joint affected by collision.

While having many benefits, aforementioned and various other reactions to collision can be implemented only on certain models of robots, and in ways which might not guarantee safety. Since focus of this algorithm is on implementation of safe reaction on collision, and in a way which is easy to implement on various brands and generations of robots without any external equipment, other reactions are out of the scope.

3.1.4. Experimental Validation and Discussion

Implemented algorithm has been tested on Denso VP-6242 robot with RC8 controller. All the functions and procedures are implemented in robot's controller using Wincaps III environment, which is standard programming software for this brand of robots. Demonstration of collision detection can be seen on https://youtu.be/nO4i5bA0gIU.

The first phase of testing was performed offline, on a PC, to test the performance of rules for collision identification. First, more than hundred sequences of execution of the repetitive routine without collision were recorded. After that, similar thing was done, but for sequences in which at least one collision has occurred in each period of execution. During acquisition of the sequences with collisions, it was attempted to generate various types and intensities of collision, by pushing, pulling, gradually increasing force, or making short time impacts on the robot.

Some of sequences without collision were used to generate reference torque limits and set tolerances. For each of the calculated reference torque limits and tolerances, there were two cycles

of testing. The first cycle of testing was with remaining sequences without collision. The aim of this testing was to count number of false collision identifications caused primarily by variations in number of samples per period. The result showed that if the signals used for forming upper and lower limit were shifted for five samples compared to original signal, false identification happened in less than 1%. However, if the signals were shifted for nine samples, there was no false identification. Even though significant shifts of recorded signals were noticed, triggering signal, described earlier, has ensured that signals stay in phase, and shifted references have ensured that only irregular sequences can deviate from them.

The second cycle of offline testing for each set of limits and tolerances was conducted with sequences with collision. The aim was to determine if collisions can occur without detection. For this testing, limits were formed from signals shifted for nine samples compared to the original signal. Therefore, primary concern was in detecting collisions in areas where acceleration changes sign, where first rule is less sensitive, and where difference in subsequent samples is more restrictive detection rule. Test results have also shown that combination of two rules has resulted in only one false identification as a result of first rule, and zero false identifications caused by second rule. At the same time, zero real impacts which exceed set tolerances were ignored.

The second phase of testing was primarily focused on verifying the reliability of collision detection. The detection was tested for the entire robot, as well as for each joint individually. Individual testing for each joint was performed by increasing the thresholds for all other joints. In this manner, overall execution of commands was not modified in any way, and the program itself operated in normal conditions, while only being sensitive to desired joints.

For each joint, it was attempted to cause various types of collisions, with different timing, intensity, duration, and impact location. For the entire robot, identification of joints affected by impact was used in addition, in order to verify proper identification.

The robot was also left on several occasions to repeat its task for 4 h without stopping, in order to check for false collision detections. Testing on robot has only confirmed testing results from offline testing, with zero false detections, and satisfactory sensitivity to collisions. Results given on Figure 11and Figure 12 show proper identification of collisions on first three joints using first decision rule, and one occasion where only second decision rule detected collision on third joint. Collisions are detected with first sample which exceeds the limit, the movement is stopped within same cycle execution, after which the program will execute one more sampling cycle. Unfortunately, without external force sensors, the correct reaction time cannot be measured. However, it was estimated that reaction time is around 0.02 s.

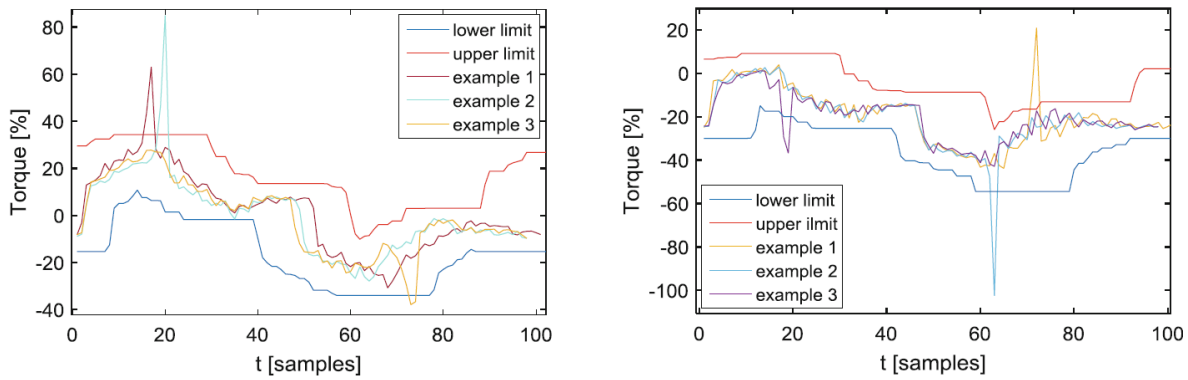

Figure 11. [13] Left graph shows three occurrences of collision detected using first rule on first axis, while right graph shows application of first rule on three examples on second axis.

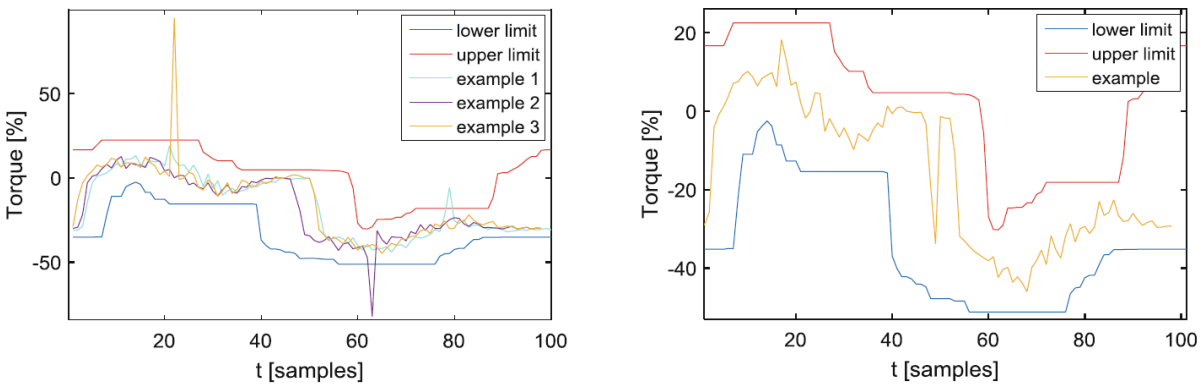

Figure 12. Three occurrences of detected collision on third axis [13] (left) using first rule and one example of collision detected with second decision rule (right).

The third phase of testing was related to convenience of use. Robot is programmed in usual manner, with the only difference at the beginning of the repeating sequence, where a trigger must be activated with a single command. All torque acquisition, processing, and comparison related to collision detection are done by service routines.

These routines run in background, in parallel with the main program, and they are started automatically by switching to mode for automatic execution. To ensure proper execution, the algorithm related routines are protected from unauthorized persons. During setup phase, the user needs to perform only four simple steps. First, the user chooses appropriate procedure by setting corresponding variable values on the robot, after which recording of the torques starts automatically. When this phase is over, the second routine will automatically start, and determine duration of one period of the repetitive movement. The second step for the user is to transfer recorded measurements and period duration to PC, where a routine automatically processes signals, and calculates referent torque limit values, as well as tolerances. The third step for the user is to transfer calculated values to robot's internal memory, after which PC can be disconnected. The fourth and final step of the user is to run the algorithm in collision detection mode by setting predetermined value of the variable.

Testing has shown that, although the algorithm has good performance and sensitivity, it is limited by the processing capabilities of the controller and lack of even sampling which would make it even better. The implementation of the synchronization trigger signal is of crucial importance for the proper functioning over extended time periods, and it has proven itself to be a simple, yet effective solution. The implemented safety measures and implementation simplicity have certainly increased the likelihood of proper setup and use, preventing misuse or serious consequences for the operator.

3.2. Integration on a computer

Previous related work [13], [14] has shown that it is possible to design a collision detection algorithm which has satisfying performance and which is simple enough to be implemented on the robot controller itself. The algorithm requires connection to the computer only in the setup phase, when signal limits are determined, and it can operate independently in regular use. However, the main drawback of such algorithm is that the computer needs to be connected every time the movement of the robot is modified, in order to determine new limits. Moreover, additional tests have also shown the possibility that the algorithm's performance can deteriorate over longer operation times. The reason for the deterioration can be found in small changes of the dynamic parameters of the robot during long periods of execution, which then affect profiles of the measured signal. Although such changes do not affect the safety aspect of the system, they may occasionally cause false alarms since the algorithm is not flexible enough to adapt to them.

The motivation for this section is to design an efficient collision detection system that can be used with various brands of existing industrial robots without any external sensors or alterations to robot's control algorithms or mechanical design. In order to achieve better performance and reliability, collision detection algorithm itself is implemented on a PC with permanent connection with the robot's controller. The following sections give special attention towards solving implementation issues along with experimental validation on two robots from different brands. By developing a collision detection algorithm for industrial robots with no such capabilities, the intention is to minimize the risk and severity of potential injury to human workers, making a major step towards fulfilling conditions for safety and coexistence.

3.2.1. Principal Design

This section is dedicated to presenting the principle on which this detection algorithm is based on. As mentioned earlier, due to the increased complexity of the algorithm presented in this section, the possibility of its full integration on robot's controller is reduced for many brands because of their closed architecture and lack of interface [12]. Therefore, it was decided to explore benefits and performance of more complex collision detection algorithms by implementing them on a PC which has permanent connection to the robot. In order to preserve versatility and brand-independent applicability, it was decided to use Ethernet TCP/IP connection, since it is a standardized and widely available communication connection type.

The overall structure and behaviour of the collision detection system is designed to be simple in order to minimize reaction time and load on the robot side. Following descriptions aim to illustrate the principal workings of the algorithm on a basis of a typical measurement acquisition/collision detection cycle.

The robot performs its intended repetitive task. The background parallel procedure on the robot sends measurements of values of interest for each joint to the algorithm operating on a PC. Each measurement sent by the robot contains only one sample of current/torque for each robot joint.

The algorithm on the PC receives the measurement and uses it in accordance with one of the two scenarios. If the reference is being formed, then the algorithm records a certain number of measurements and uses them as the reference sequence. Otherwise, if the algorithm is actively detecting collisions, then the measurements from the robot are used to form a measurement vector. The measurement vector contains most recent and several preceding successive measurements. Procedure on the PC then matches the measurement vector to the corresponding section of the reference sequence. Using matched signals, collision detection rules decide whether the collision occurred on any of the robot joints.

A signal indicating whether a collision was detected is sent from the PC back to the robot. The robot stops its movement if the collision was detected. Otherwise, the robot sends new measurement to the PC and entire cycle repeats. In case of failed communication, the robot will stop as a safety precaution.

Within the described structure of the collision detection algorithm, there are a few matters implemented on the PC which deserve additional description.

The first aspect is related to the choice of reference signal. The background on which the algorithm relies on is the reality that majority of robot tasks are repetitive. With robot operation repeating in cycles, the intuitive choice of reference would be to exploit this fact. Signal values from more than one cycle of robot operation can be recorded for each joint and used as a reference signal. As mentioned earlier, if the reference signal is acquired during real operation of robot performing its tasks, it brings important benefits. Most importantly, it incorporates the nominal robot dynamics during the desired movement. In addition to that, external forces occurring during payload manipulation, assembly or processing are intrinsically included in the recorded signal. Finally, the signal includes potential influences of robot dynamics related to changes in robot behaviour over time. It is important that the reference signal contains more than one cycle of recorded repetitive robot movement. This ensures that the transition from one cycle of the repetitive movement to another is also included in the reference. Another important, although intuitive note is that no collision should occur during recording the reference signal. To detect collisions, reference signal is compared with measurement vector which is periodically updated from the robot. The measurement vector contains relevant signal values for each joint of the robot for several time instants. The vector is formed of adequate number of signal samples and updated with new measurements with first-in-first-out logic. Therefore, the measurement vector contains measurements from a fraction of the repetitive movement of the robot.

As mentioned earlier, the reference sequence contains signal values more than one cycle of repetitive movement. In order to compare it with measurement vector, it is needed to find a section of reference signal which corresponds to the content of measurement vector. As described in previous sections, the sampling frequencies of reference signal and measurement vector do not necessarily need to be identical. To overcome this situation, a modification of Dynamic Time Warping Algorithm (mDTW) was developed. It allows online matching of reference sequence and measurement vector and enables application of collision detection rules.

Collisions of robot are identified based on two detection rules. Matched signals are compared, and if either of the two detection rules is triggered, a signal is immediately sent to the robot to stop the movement. Thresholds for decision rules are set based on statistical analysis of the measurements without collisions.

With regards to the collision reaction strategy, various options are available. Strategies include stopping the robot, performing reflex motion away from the direction of impact, going to zero-gravity compliant mode or attempting partial completion of the task using redundant joints while trying to avoid further contact [9], [44], [4], [19], [10], [12]. Although fully recognizing the benefit of refined collision reaction strategies, the focusof further elaboration is more on reliable detection and its applicability across various brands. As such, the algorithm proposed in this section employs stopping upon detected impact as the simplest reaction strategy which is also available on vast majority of robot brands and generations.

Although the developed mDTW presented in this section offers numerous benefits, there are some limitations. A potential drawback of using mDTW is that it effectively limits the number of samples which can be used for the reference sequence and measurement vector. While there are no theoretical limitations of the proposed method related to the length of compared signals, there are practical issues related to reaction time. For a collision detection, the reaction time needs to be as short as possible. For mDTW, processing time depends directly on the number of samples in each of the compared signals. Nevertheless, there are effective ways of managing this issue. Reducing sampling frequency can reduce the number of samples in reference signal. If handled responsibly, this solution can reduce the computing time without loss of detection quality thanks to the nature of mDTW. Additional solutions include parallel processing for different axis of the robot, as well as changing hardware platform on which the algorithm is implemented. Although not affecting the computation time, switching to a faster communication protocol can speed up the overall reaction time.

It is important to emphasize that the presented algorithm can work with current measurements and torque measurements/estimates alike, making it highly versatile. The reason for opting to present results with torque estimates, rather than current measurements, is purely to make them easier to comprehend. Torques have a more physical manifestation, making them more understandable. Additionally, measurements of joint current provided by Denso are in absolute values, which means that graphs would not clearly illustrate the nature of the movement.

3.2.2. Modified Dynamic Time Warping

One of main contributions presented in this section is a modification of the Dynamic Time Warping (DTW) algorithm. Similar to DTW, the proposed modification enables non-linear time axis transformations to perform optimal matching of two signals. However, unlike DTW, the proposed modification also enables optimal matching of one sequence with most similar section of the other sequence, rather than comparing the two entire signals. The modification enables use of DTW principles in online applications such as collision detection. Although there are variants of the DTW which allow matching of certain sections of one signal to another signal [46], [47], no other existing modification allows combination of such traits at the same time.

In early phases of algorithm development, various techniques were considered in order to solve the issue of variability and inconsistencies of measurement signal sampling. Methods such as EDIT [48] distances, was one of the first to be analysed, since it allowed for different lengths of compared signals and some time-domain distortion. Regardless of its potential, it was abandoned because of two main reasons. The most important drawback was that it allowed some values to be deleted and/or added, which meant that values indicating collision could be missed due to nature of the algorithm. This issue remained present to a similar extent also with similar techniques, such as Hirschberg's algorithm [49] and Approximate string matching [48]. The second reason for abandoning this approach was that it was intended for a finite set of values of compared signals, such as in string comparison. This would require significant changes to the algorithm or introduce value quantization related heuristics. Methods such as Partial Curve Mapping [50] and Discrete Fréchet Distance [50] were also analysed due to their versatility. Nevertheless, the key drawback of such methods was that they were intended for parametric curve comparison, rather than time series.

Dynamic Time Warping (DTW) [51] is a well-known method used to match signals with different lengths and similar shape. It is widely considered as a reference for shape-based matching of the signals in time series [52]. Consequently, it can also be used as a good measure of similarity of two signals. It allows for contraction and/or dilatation of a signal in the time domain, and uses matrices to find sequences of best matching pairs of samples from two signals a and b. By definition, the matrix d based on which optimal pairs are determined is formed using rules (1)-(4):

$$d(1,1) = |a(1) - b(1)|, i = 1, j = 1 \quad (1)$$

$$d(1,j) = |a(1) - b(j)| + d(1,j-1), i = 1, 1 < j < n \quad (2)$$

$$d(i,1) = |a(i) - b(1)| + d(i-1,1), 1 < i \leq m, j = 1 \quad (3)$$

$$d(i,j) = |a(i) - b(j)| + min(d(i-1,j-1), d(i,j-1), d(i-1,j)), 1 < i \leq m, 1 < j \leq n \quad (4)$$

In the aforementioned expressions, m and n stand for lengths of the first and the second signal. In essence, rule (4) implies that $d(i,j)$ carries the information about the minimal possible accumulated absolute difference of all samples preceding and including i-th sample of signal a and j-th sample of signal b. To achieve a minimal absolute difference, any sample of signal a can be matched with multiple samples of signal b, as well as the opposite. This type of matching effectively causes apparent signal contraction or dilatation. Rules (1)-(3) are only special cases of rule (4), used to fill the starting field, first row and first column of matrix d. For the starting point,

there are no preceding samples, so there is no other way to compute the field value, but the absolute difference of first samples of both signals. For the first row, value of preceding element of the first row is added to absolute difference of the first sample of signal *a* and *j*-th sample of signal *b*. Analogue logic is applicable for rule (3) and forming values of the first column of matrix d. Effectively, rules (1)-(4) force the algorithm to start from the first element of the matrix d because it has the lowest possible value.

Best matching pairs within the matrix *d* are found using rules described with (5)-(7), starting from the last sample of both sequences *d*(*m*,*n*).

$$min(d(i-1,j-1),d(i,j-1),d(i-1,j)),i \neq 1,j \neq 1 \quad (5)$$

$$d(i,j-1),i=1,j \neq 1 \quad (6)$$

$$d(i-1,j),i \neq 1,j=1 \quad (7)$$

Rule (5) implies that the search for the optimally matched pairs starts from the element *d*(*m*,*n*), and looks for the minimal value among all neighbouring preceding elements. The search does not allow going back, preserving the order of samples and therefore the causality in matching. In case first row or first column is reached during the search, the search is directed towards the first element, belonging to beginning of both signals, according to rules (6) and (7). The matching is finished when the *d*(1,1) is reached.

It is possible to notice that the search supposes that the beginning and the end of both signals are common. One more important property of DTW is that it does not allow skipping of any samples. The two aforementioned observations lead to the conclusion that the contents, or shapes, of the compared signals needs to be similar in order to successfully perform matching. In other words, it is not possible to take a subsequence of one signal and find the best matching section of the other signal.

The fact that DTW can only be used to match sequences with same shape i.e., content means that it cannot be used for online collision detection, as it would only be able to detect collision after the entire sequence has been recorded. Solutions proposed and described in [53] offer increase in speed and a modified measure of dissimilarity of compared signals. However, they are still not suitable for application in collision detection, as they presume the same starting and ending point of compared signals. Some research has also gone into reducing calculation speed by implementation on different platforms and some early pruning [54], but this type of research is not a subject of this research.

Some modifications of DTW were proposed to enable partial matching of signals. However, they require either beginning or the end of the signal to match [46], [47], or allow skipping some samples, such as in Minimal Variance Matching [55]. In collision detection application, skipping some samples is unacceptable, because the algorithm could potentially skip samples which indicate collision. On the other side, it is possible to implement an algorithm which requires only matching of beginnings of the two signals. In this case, the signal matching would be re-initiated with a trigger signal at the beginning of each new cycle of robot's movement. However, as length of measurement increases over time, so does the matrix, and the time needed for matching, making the algorithm increasingly ineffective and unreliable over duration of one movement cycle. Another solution [46], [47], which allows a partial matching of signals to occur does so effectively, but allows both signals to be trimmed from either end, which is not acceptable in application in collision detection as those sections might contain important samples indicating collision.

In addition to aforementioned issues related to implementation of DTW in collision detection, there are also some other aspects to consider. One of the biggest issues related to DTW is the computation time which depends on the lengths of matched signals. While it is not an issue in offline applications, processing time is very important for timely collision detection, and it needs to be considered.

The presented modification of DTW enables matching a subsequence of any signal with corresponding section of the entire sequence. In particular case, it enables matching of set of measurements from the robot with previously acquired reference sequence of current/torque. The impact is that this modification enables online implementation of DTW - mDTW, and thus implies that mDTW can be used in collision detection. Additional benefit originates from the fact that one of the signals can be significantly shorter than the other. This fact offers possibility for significant reduction in computing time, making it even more suitable for collision detection application.

The proposed modification modifies the rules (1)-(4) for forming the matrix *d* with rules (8)-(11).

$$d(1,j) = |a(1) - b(j)| + min(|a(2) - b(j+1)|, |a(2) - b(j)|, |a(1) - b(j+1)|), i = 1, j \neq n \quad (8)$$

$$d(1,n) = |a(1) - b(n)| + |a(2) - b(n)|, i = 1, j = n \quad (9)$$

$$d(i,1) = |a(i) - b(1)| + d(i-1,1), 1 < i \leq m, j = 1 \quad (10)$$

$$d(i,j) = |a(i) - b(j)| + min(d(i-1,j-1), d(i,j-1), d(i-1,j)), 1 < i \leq m, 1 < j \leq n \quad (11)$$

Rules (8)-(11) introduce difference in calculating matrix *d*. Rule (8) implies that all elements in the first row of matrix *d* are calculated based on absolute difference between the first sample of signal *a* and the *j*-th sample of signal *b* increased by a minimal absolute difference of any of the neighbouring succeeding samples. This ensures causality as well as that all elements in the first row have a chance to be the end point of the search. The search no longer has to end with *d*(1,1), but rather any element in the first row. This means that signal *a*, which is supposedly subsequence of signal *b*, can start at any sample of signal *b*. Rule (9) is a special case of rule (8) which ensures the consistency for the search algorithm. Rules (10) and (11) are the same as rules (3) and (4) respectively, and they ensure causality.

The search for optimal pairs starts from the element with minimal value in last row of the matrix, min(*d*(*m*,j)), 1≤j≤*n*. The search continues with finding the minimum of preceding values as described with (12)-(13), and stops immediately when the first row of the matrix is reached.

$$min(d(i-1,j-1), d(i,j-1), d(i-1,j)), i \neq 1, j \neq 1 \quad (12)$$

$$d(i-1,j), i \neq 1, j = 1 \quad (13)$$

Since values increase monotonically with an increase of row and/or column number, the algorithm inherently ensures that the minimal value in the last row is the optimal starting point for the search. Modifications (8)-(11) ensure that each field in every row and/or column but the first row includes information about the minimal sum of deviations before it, making sure that optimal path to that particular point can be traced back to the first row.

The proposed mDTW method enables effective application in collision detection. A set of *m* consecutive measurements (measurement vector) with first-in-first-out logic can be compared to a subsequence of the reference sequence with *n* values. In this thesis, a reference signal to which all other signals are compared is called the reference sequence. The reference sequence is compared with a signal called measurement vector, as in a real application, it represents a set of *m* consecutive measurements received from the robot. The reference sequence remains constant during the collision detection. Measurement vector is formed from m consecutive samples which are updated in each detection cycle. The movement of the robot is periodical, and as a consequence of that, so are the measurement vectors, which represent a section of repetitive movement. This ensures that signals do not change without bound over time in nominal operation of the robot. The only time the vector can deviate is when a collision occurs, which is not a regular occasion.

For simulation purposes, measurement vector is formed using a window of *m* samples moving across signal referred to as measurement sequence. Section of the reference sequence to which the measurement vector was matched is called reference subsequence. Graphs presented on Figure 13 show values of *d* matrix of mDTW, measurement vector matched with reference subsequence, highlighted section of reference sequence representing reference subsequence and section of the measurement sequence from which the measurement vector was formed. In a real application, from the implementation point of view, initial set of measurements can be formed as a vector of *m* identical values of the first obtained measurement and filled with new values as they are received from the robot. At the beginning, until *m* samples are received from the robot, any identical values will be compressed and matched with a single value from the reference sequence, and all following samples will be matched with their best matching values from the reference sequence, as it is illustrated on Figure 14.

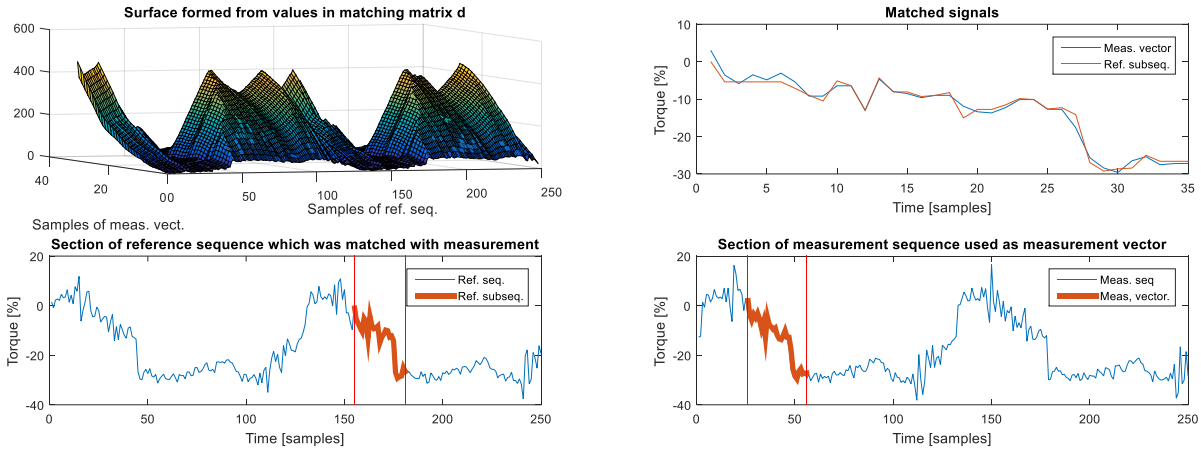

Figure 13. Signals matching performed by modified DTW (mDTW) [14]. (upper left) Values of matrix used for optimal matching. Optimally matched pairs of signals correspond to the lowest sections of the surface. (upper right) Two matched signals - blue signal corresponds to the section of the reference signal, while the red represents the measurement vector received from the robot. (lower left) The signal which shows from which section of a measurement sequence the measurement vector was taken. (lower right) The signal shows which section of the reference sequence was matched with the measurement vector.

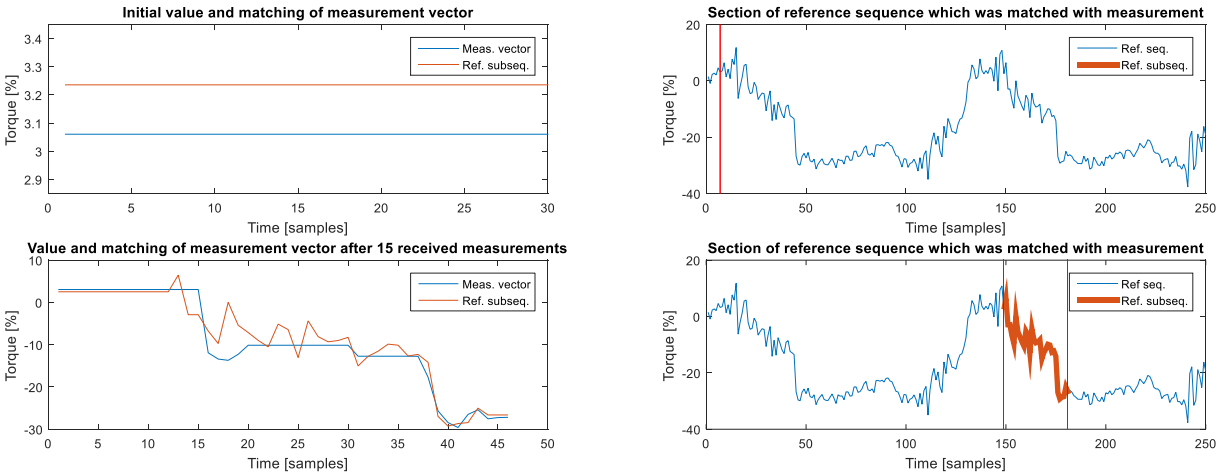

Figure 14. Two instants during formation of the 30 samples long measurement vector [14]. (upper left) Initially, all values in the measurement vector are the same and equal to the first measured value received from the robot, Therefore, they are matched with single value of the reference sequence. (upper right) Point of the reference sequence paired with the measurement vector. (lower left) Measurement vector shown in blue colour is partially updated with new values upon receiving them from the robot, while initial values are still present at the beginning of the vector. The new vector is matched with the most similar section of the reference signal, shown in red after each new measurement is received. Although measurement vector has 30 samples, the matching process has dilated it to 46 samples to perform optimal matching. (lower right) The section of the reference sequence to which the updated measurement vector was matched.

Similar to the solution described in [53], [52], the presented modification limits the signal lengths, and consequently the computational complexity, by using constant lengths of the reference sequence and of the measurement vector. However, the proposed modification allows for two key advantages which are important for collision detection application. The first advantage is the possibility to perform matching one signal to a most similar part of the other signal, which enables online application in collision detection. The second advantage is that, unlike [55], mDTW method does not allow skipping of samples, which prevents missing a collision. With all aforementioned said, it is important to note that the computation time mDTW still depends on the lengths of signals. However, this property, inherited from the original DTW, is much less prominent. Recently published work [52] also presents techniques which limit computational requirements, such as incremental computation and restrictions of search region. While incremental computation is a big step in reducing the computational complexity, it is not applicable in the specific application of collision detection without modifications, since it assumes that the beginning and the end of compared signals is the same.

3.2.3. Decision Rules and Thresholds for Collision Detection

In the proposed approach, decision-making rules which determine whether the collision has occurred or not, were derived from statistical analysis of the signal. As a result of the analysis of signals previously matched with mDTW, two complementary decision-making rules were designed.

The first rule was derived from the most intuitive feature of signal matching after application of mDTW, and that is the absolute deviation of a sample from its pair. For each joint of the robot, statistical analysis was performed on a set of more than 500 cycles of repetitive motion in order to determine the maximum permissible deviations of signals without collision. The set limits also determine what will be the minimum detectable collision. The limits are set as the first integer value higher than the 3σ value calculated from samples which did not belong to a collision, similarly as implemented in [12]. After the absolute values of permissible deviations were set for each joint, the rule was applied as a first and the faster form of detection. During matching of sequences samples, the search for optimal pairs is started from the last row of the matrix, which means that the most recent measurement is matched first. If the minimal value of deviation for the most recent measurement sample is bigger than the set threshold for the corresponding joint, the collision is indicated, and the robot is stopped, with minimal reaction time. Two examples demonstrating the first rule being inactive and active are shown in Figure 15.

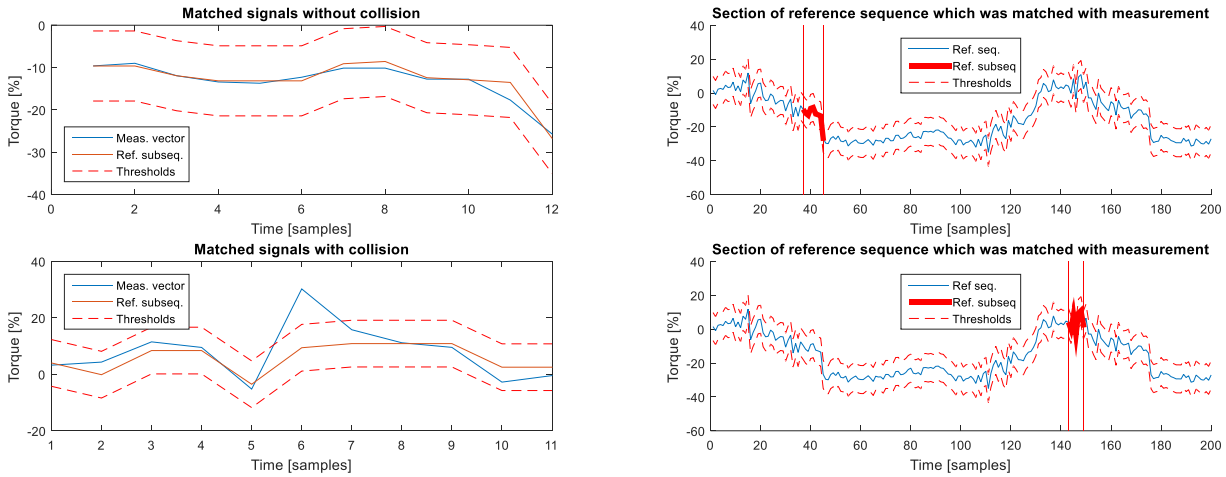

Figure 15. Matched signals on 2nd axis of the robot [14]. (upper left) Section of the reference signal matched with the measurement vector without collisions as well as signal thresholds imposed by the first rule. (upper right) Subsequence of the reference signal to which the measurement from the upper left picture vector was paired and thresholds set by the first rule. (lower left) A measurement vector which was paired with the most similar subsequence of the reference signal. The sample belonging to the collision is over the upper threshold. (lower right) Subsequence of the reference signal which was most similar to the measurement vector.

The second decision rule complements the first one and increases the reliability of collision detection. It is best illustrated when relative dependency of matched pairs is observed, as depicted on Figure 16. Since matched pairs have similar values, their dependency on sig1-sig2 plane is approximately linear. The direction of the line fitted to paired samples approximately corresponds to the direction of main Eigen vector of the covariance matrix of all matched samples. The direction of the second Eigen vector corresponds to the dissipation of samples from the main Eigen vector direction. Consequently, intensity of the second vector, or the second Eigen value, can be used as a reliable measure of how well the samples have been matched. Samples which have not been matched properly, will contribute to the noticeable increase of the second Eigen value even if the absolute difference is not significant. By performing a statistical analysis of the signals without collision for each joint, it is possible to determine limits for this decision-making rule. Thresholds for this rule were derived from the statistical analysis and set to values 20% higher than the highest second Eigen value during operation without collision. Unlike the first rule, which is related to the absolute difference between last pair of matched samples, the second rule is related to the square value of the maximum difference between any pair of matched samples. This means that the second rule has a more restrictive threshold, meaning it is more sensitive to collisions. This rule also observes the entire sequence, so any potential collision has potential to be visible for at least the number of cycles equal to the length of the measurement vector, drastically reducing the possibility of missed collision.

The two described rules observe different aspects of the matched signals and complement each other to drastically reduce the chance of a missed collision. The first rule focuses on only the most recent matched sample in order to react promptly before the complete matching is performed. It is therefore especially sensitive to higher intensity collisions. The duration of the collision does not affect the quality of detection for the first rule, since it only observes the last matched sample. The second rule is more sensitive to short impacts, and transients from nominal behaviour to collision. While each rule independently has its own benefits, the highest performance in collision detection is reached when they are used simultaneously.

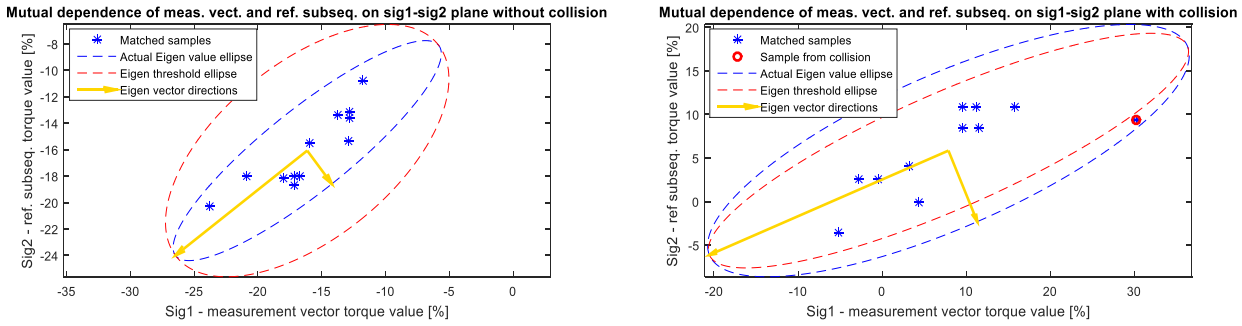

Figure 16. Illustration of the Eigen values rule [14]. (left) Approximately linear distribution of paired samples without collision on sig1-sig2 plane. Higher intensity Eigen vector approximately corresponds to the line fitted to the measurements. The lower intensity Eigen vector is a measure of dissipation of sample pairs along direction orthogonal to the higher intensity vector, and it is used to identify collisions in the second rule. (right) Sig1-sig2 plane with indicated collision marked with the red circle which is out of the threshold indicated with a red dashed line. The collision sample influenced the orientation of Eigen vectors and increased intensity of the lower intensity Eigen vector.

3.2.4. Experimental Results and Discussion

The performance of the algorithm was tested in two main phases. The first phase of testing was performed offline, on a PC, with a large number of recorded measurements. The goal of offline testing was to validate mDTW, decision rules as well as communication and interfacing programs on the part of the collision detection algorithm implemented on the PC. The second phase of testing was done with a real robot with a running collision detection algorithm, and its goal was to experimentally validate the performance of the collision detection algorithm.

Current-based torque estimations can be obtained in majority of robots directly, and the presented algorithm is analysed and tested based on the torque estimations. The results are validated for two different robot brands (Denso and ABB) using current-based torque estimation. Since torque estimates are obtained directly from the robot, as if they were real measurements, they will be referred to as torque estimations. For Denso robot, torque measurements range from -100% to 100% of permissible torque in each joint, and all graphs based on measurements on this brand will be presented in these limits. Graphs from ABB robot are shown with real torque values, and all measurements and results related to this brand will be presented in absolute values.

During implementation, two routines were designed for part of the algorithm implemented on the PC. The first routine is very simple, since its only task is to record the movement of the robot with high sampling frequency. The reason for this is to capture the shape of signal from the movement cycle with high level of details. The higher sampling frequency is a reason why the reference signal is also recorded using this routine. During recording of the reference signal, no collisions should be allowed.

Set of sequences used in offline testing was recorded with the first routine to provide greater flexibility for testing. Additionally, sequences recorded this way enable better understanding the influence of different sampling times to interpretation of signal features and detection of collisions. Some sequences were recorded with and some without collisions, so that different aspects of algorithm performance can be evaluated.

The second routine is used for online collision detection, and it is more complex. Due to its complexity, it requires a longer execution time. The robot samples its movement once per cycle of PC program execution, while executing collision detection routine. Therefore, the sampling frequency of the second routine will be lower than with the first routine, which is used for reference acquisition only.

Sequences used in the online part of testing were recorded with the second routine, which is used for collision detection in the online loop. Therefore, they realistically illustrate the quality of online collision detection. Similar to the offline part of testing some of the used sequences were recorded with and some without collision, in order to verify the performance of the algorithm,

The discussion part of this section summarises results from both offline and online testing. It also comments the benefits and drawbacks of the algorithm concept as well as of decision rules.

Offline Testing

The presented collision detection algorithm relies on optimal matching of a reference sequence with a recorded vector of measurements. Therefore, the first part of offline testing was performed on mDTW to check for any mismatching of samples. The target is to receive a vector of m measurements from the robot and match it optimally with the most similar subsequence of the previously recorded reference sequence of torque signal. To perform this type of test, reference and measurement sequences were both emulated – selected from the previously recorded sequences.

To form a measurement sequence, a sequence was chosen from the set of previously recorded sequences. In order to emulate a measurement vector, a window of m samples was taken from the measurement sequence. This window was selected by moving for 1 sample along the measurement sequence in every cycle of matching, just as the measurement vector would be updated with one new measurement in each new cycle in realistic situation. Similarly, for reference

sequence, any sequence can be chosen from a set of previously recorded sequences which do not contain any collisions.

In order to verify the concept of the mDTW, at the beginning, the reference and measurement sequences were the same signal. For the same sequences, the zero matching error proved the concept without exceptions. In part of the experiment that followed, the same test was repeated by using a different measurement sequence. The result of the second part of mDTW testing proved the optimal matching regardless of whether the measurement sequence was taken from the set of measurements recorded with or without collision. Figure 17 depicts two examples of mDTW testing with different measurement sequences – one with collision and one without collision.

Another phase of mDTW testing was conducted to verify the performance of matching signals recorded with different sampling time. The measurement sequence was re-sampled with up to 3 times higher and lower frequency, as well as non-integer multiples of original frequencies. Results have shown that the matching of samples was performed optimally, regardless of the differences between sampling times. An example of testing with various sampling times is shown in Figure 18. The results have also shown that the matching error increased with the difference in sampling times. However, these results were in accordance with expectations, since significantly different sampling frequencies mean that one sample of one sequence needs to be paired with several samples of the other sequence, all of which have different values, contributing to the cumulative matching error.

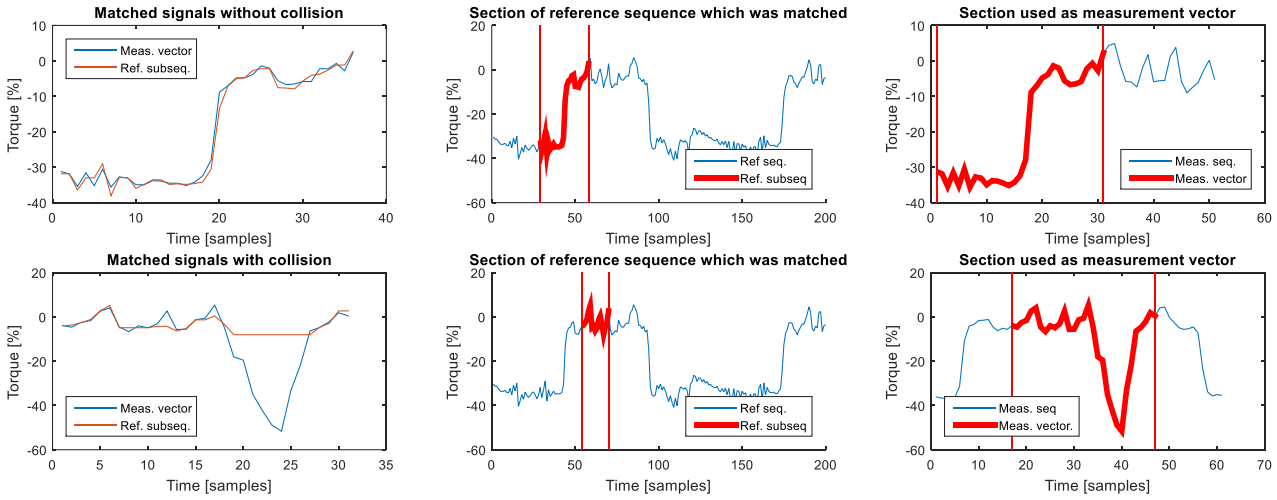

Figure 17. Testing of modified DTW, example on the 3rd axis [14]. (upper left) Measurement vector matched to the subsequence of the reference sequence. (upper middle) Matched subsequence of the reference sequence. (upper right) Portion of measurement sequence from which the measurement vector was formed. (lower left) Measurement vector taken from a sequence with collision and mached with most similar subsequence of the reference equence. (lower middle) Subsequence of reference sequence which was matched. (lower right) Portion of the measurement sequence used to form the measurement vector.

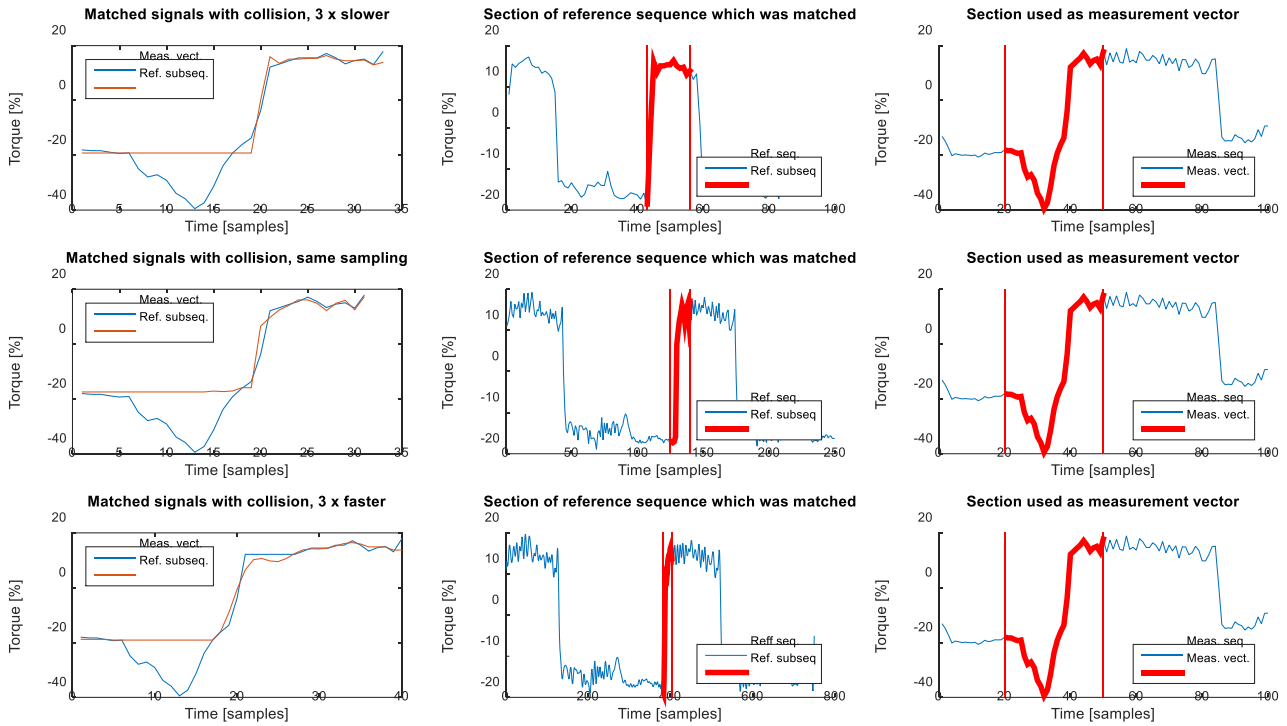

Figure 18. Influence of sampling frequency on the quality of matching on the 4th axis [14]. (upper row) Reference sequence sampled at 3 times lower frequency compared to measurement sequence. (middle row) Reference sequence sampled at the same frequency as measurement sequence. (lower row) Reference sequence sampled at 3 times higher frequency compared to measurement sequence. (left column) Matched samples of measurement vector and reference subsequence with different sampling time. (middle column) Subsequence of the reference sequence to which the measurement vector was matched. (right column) Same section of the measurement sequence was used to form the measurement vector in all three cases.

The second part of offline testing was performed to validate the two collision detection rules. The results from offline testing of individual and combined use of collision detection rules are shown in Table 3. The first collision detection rule testing was to check the rule which considers absolute value of the difference between matched samples. As mentioned earlier, the purpose of this decision rule is to make prompt identifications of collisions. The measurement vector was obtained in a previously described manner, and the deviation of the last matched pair was observed. The set of measurement sequences used in this phase of testing was not the same set which was used for statistical analysis of the signal. Based on the previous statistical analysis, the limit for each axis of the robot was set to a value as described in the previous section. With limits set by the statistics, this decision rule has successfully detected all major collisions. One detection example using the first rule is shown in Figure 19, while the statistical analysis of the first detection rule can be seen in Table 3.

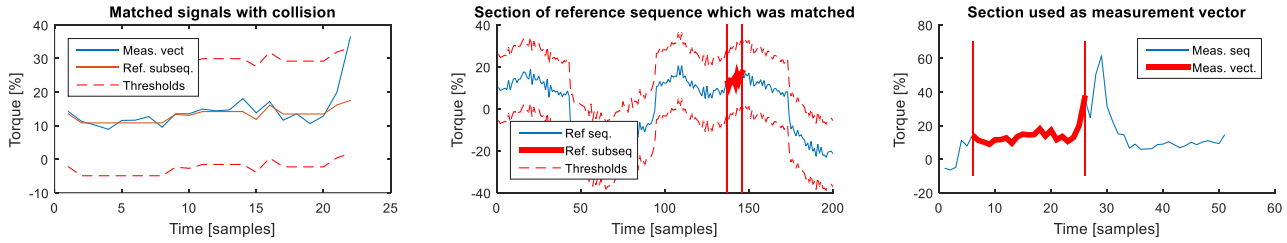

Figure 19. Example of the first rule application on the 1st robot axis [14]. (left) Sample of measurement vector has crossed the threshold, triggering the collision detection. (middle) Subsequence of the reference to which the measurement vector was matched. (right) Portion of the measurement sequence used to form the measurement vector.

Table 3 Statistical results from the offline testing. [14]

	Collision not Detected			Collision Detected		
	Rule 1	Rule 2	Combined	Rule 1	Rule 2	Combined
Sequences without collision	99.73 %	98.92 %	98.66 %	0.27 %	1.07 %	1.34 %
Sequences with collision	5.63 %	17.50 %	0 %	94.37 %	82.50 %	100 %

The second collision detection decision rule relies on the distribution of mutual dependence of matched samples. By observing the value of second Eigen value, the overall measure of the matching success is considered. The sensitivity of this decision rule is higher, as a single value which clearly indicates a collision contributes to the rise of Eigen value significantly. Results have shown that this algorithm has classified collisions with lower corresponding tolerance to absolute differences than the first rule. More details related to the performance of the second rule are presented in Table 3. However, the complexity of the processing behind this rule caused slightly longer times needed to identify the collision than the first rule. During several tests, it was shown that the same collision takes about 10%-30% more time than the first rule (e.g., the first rule took approximately 0.05 s for collision detection while, the second rule took 0.055 s to 0.65 s to identify the collision). Figure 20 demonstrates a particular situation in which the second decision rule managed to identify the collision one cycle before the first rule was activated due to higher sensitivity. Figure 21 shows that the collision would be detected also by the first rule, but one cycle after the second rule was activated. With regards to reaction times, this is very significant, and although such situations are not common, they are very important.

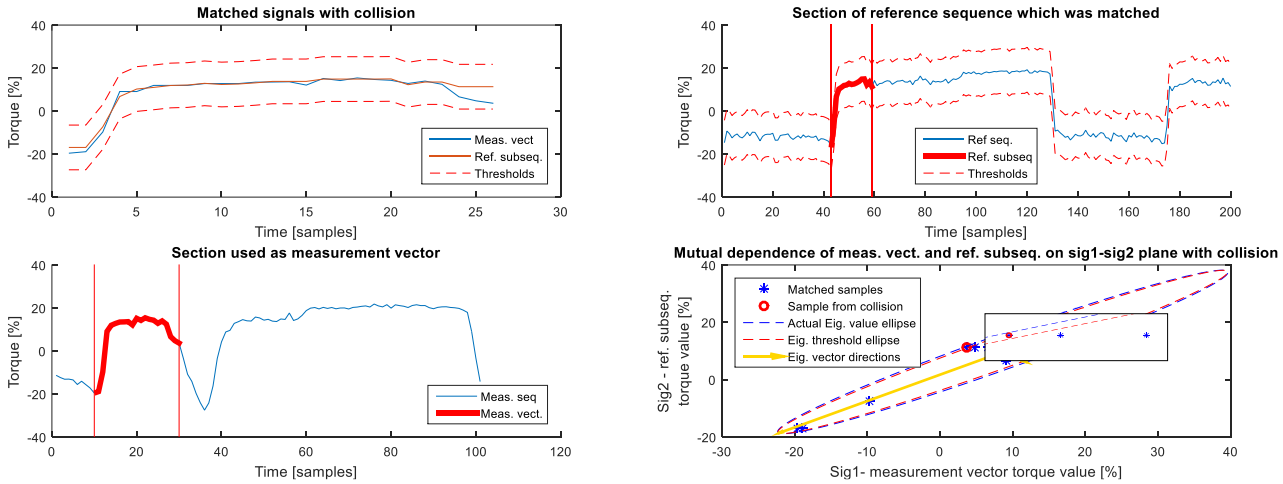

Figure 20. Example of the second rule application on 4th robot axis in moment when the first rule was not yet triggered[12]. (upper left) Sample of measurement vector has not yet crossed the threshold of first rule. (upper right) Subsequence of the reference to which the measurement vector was matched. (lower left) Portion of the measurement sequence used to form the measurement vector. (lower right) Sample after which the second rule was triggered because it was bigger than the threshold, shown in red dashed line.

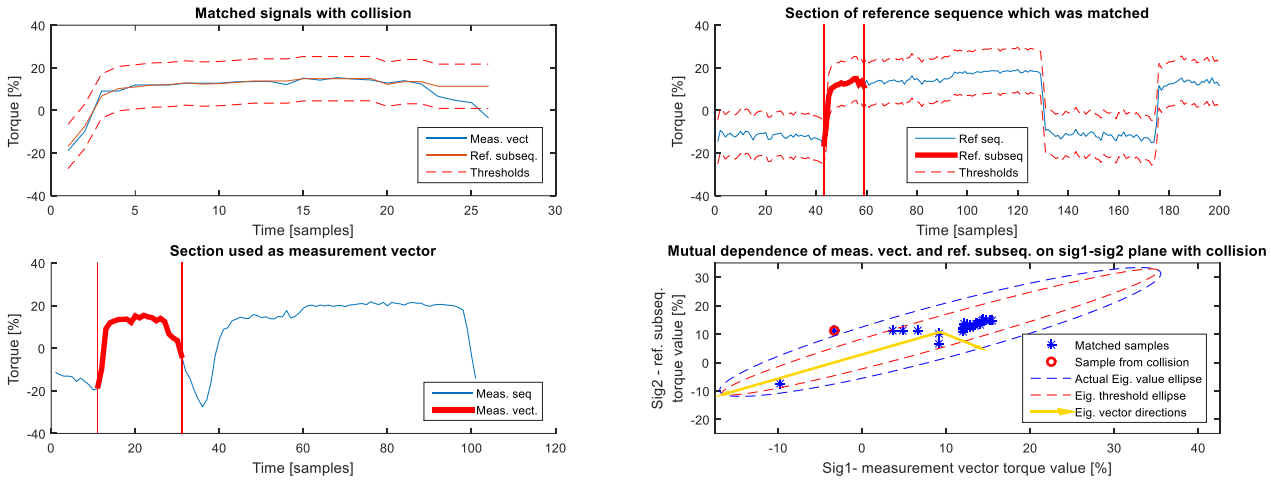

Figure 21. Example of the first rule being triggered for collision on the 4th robot axis one sample after collision was detected by second rule[12]. (upper left) Sample of measurement vector has crossed the threshold of the first rule after one sample more compared to the moment shown on Figure 20. (upper right) Subsequence of the reference to which the measurement vector was matched. (lower left) Portion of the measurement sequence used to form the measurement vector. (lower right) Samples after which the second rule was triggered because it was bigger than the threshold.

Online Testing

Online testing was conducted with 2 robots of different brands in order to verify the versatility and implementability of the algorithm. During the development phase of the algorithm, Denso VP-6242 6-axis industrial robot was used, and all tests and statistical analysis were made using measurements from this particular model. However, every aspect of online testing was additionally performed on ABB IRB120 6-axis industrial robot. In each test, the robot was connected with the PC and communication was established over Ethernet. Robots used for experimental validation are shown in Figure 22. Videos of online testing are available on https://youtu.be/gEysqZhN7Y0 for Denso and on https://youtu.be/_SXtQOEJM_I for ABB robot.

Figure 22. Robots used in experimental validation[12]. (left) Denso VP-6242 robot used of offline and online testing. (right) ABB IRB120 robot which was used in online testing.

For each robot, a robot routine was designed to start automatically after switching the robot to automatic mode of program execution and to run in background of the main robot program. The task of this routine is to send a string containing values of torques from each axis and receive a value of 0 or 1 from the PC. The values indicate whether the robot is running normally, or the collision is detected. If the collision is detected, the movement of the robot is stopped, and it remains in this state until the return value changes to 0. As a safety measure, if the connection to the PC is lost, or if the return value is not received from the robot, the routine will stop the robot immediately. All the movements as well as all other programs and functions of the robot are programmed in an unmodified manner.

The main goal of the online testing was to check the performance of the algorithm in real exploitation. In particular, three aspects were observed as the most important – absence of missed collisions detection, collision detection time and absence of false detections.

Firstly, it was important to check whether a real collision will be detected, since it is the main purpose of the algorithm. The robot was impacted with various forces, from different directions, and with different impact durations. Results from online tests are shown in Table 2. Figure 23 shows one example of a collision detected on the 4th axis of Denso robot using the first rule. Collision detection example on the 1st axis of Denso robot, in which only the second detection rule was activated one cycle before the first rule detected it, is shown on Figure 24.

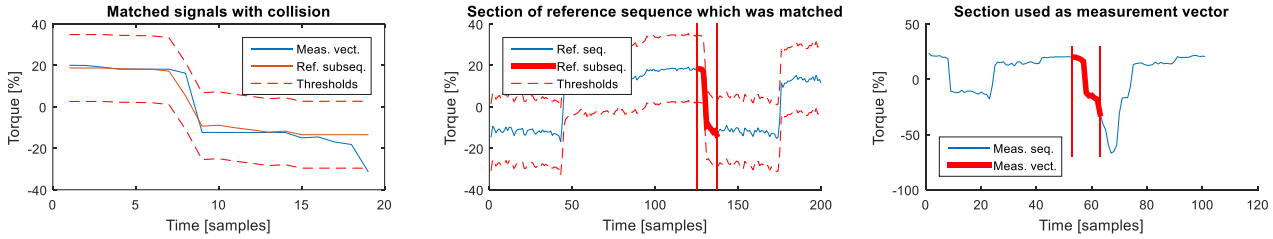

Figure 23. Results from online collision detection on the 4th axis of Denso robot. (left) collision detected using application of the first rule. (middle) Subsection of the reference sequence to which the measurement vector was matched. (right) Section of measurement sequence used as measurement vector.

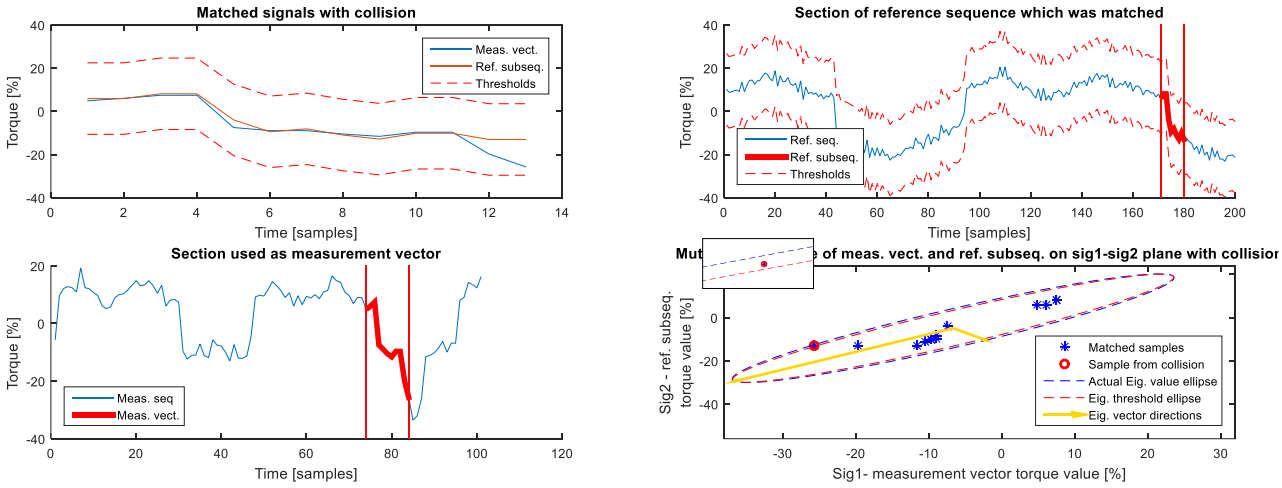

Figure 24. Results from online collision detection on Denso VP-6242 robot, in which the second rule detected the collision on 1st axis one sample before the first rule detected it[12]. (upper left) Collision is not yet detected using first rule. (upper right) Subsequence of the reference sequence to which the measurement vector was matched. (lower left) Section of measurement sequence used as measurement vector. (lower right) Sample after which the second rule detected the collision.

Experimental results from online testing on ABB robot are shown in figures below. Figure 25 shows detection of a collision on the 1st axis of ABB robot, while Figure 26 shows detection of the 3rd axis of the same robot.

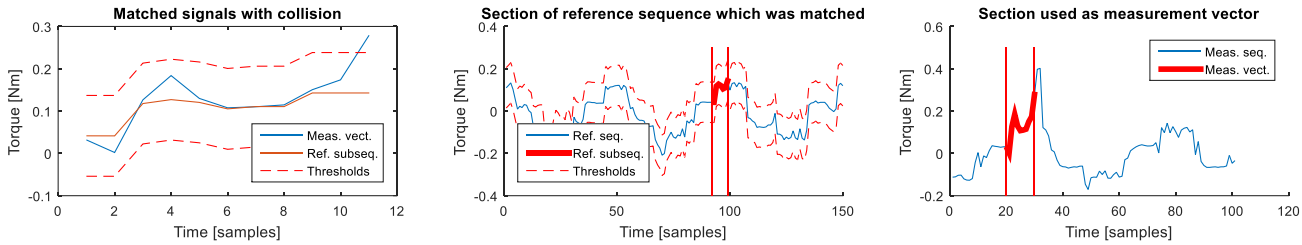

Figure 25. Results from online collision detection on ABB IRB120 robot, example from the 1st axis[12]. (left) Collision detected using application of the first rule and matched signals. (middle) Subsequence of the reference sequence to which the measurement vector was matched. (right) Section of measurement sequence used as a measurement vector.

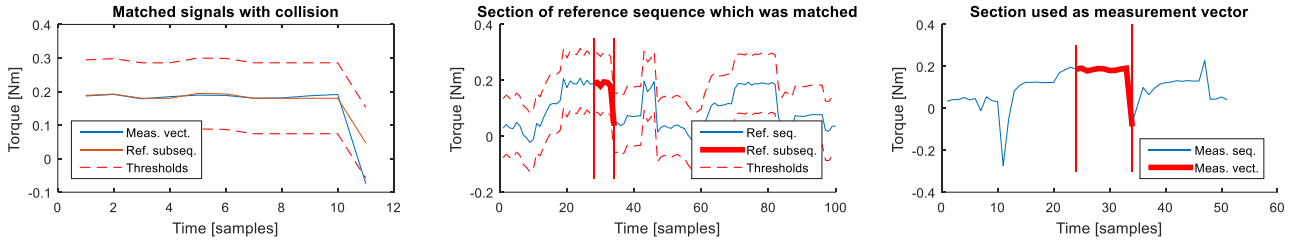

Figure 26. Results from online collision detection on ABB IRB120 robot, example on the 3rd axis [14]. (left) Collision detected using application of the first rule on the 3rd axis of the robot. (middle) Subsequence of the reference sequence to which the measurement vector was matched. (right) Section of measurement sequence used as measurement vector.

The second most important aspect is the time needed for reaction to the collision, which was tested together with the previous phase of testing. Although analytical dependence cannot be derived, and reaction time depends on the hardware (robot controller and the PC), conclusions from the extensive experimentation are listed. For a constant length of the reference sequence, the reaction time depends on the length of the measurement vector and joint on which the collision was detected. Reaction time increases with the length of the measurement vector, and with the number of joints. For example, it means that collision will be detected faster if it influenced the second axis than the fifth axis. The reason for the difference related to the affected joint, is that joints are tested in sequence one after the other, and not in parallel. Despite all variations, during online testing, the slowest reaction time was less than 0.1 s, while the fastest was 0.028 s. It is important to note that reaction time also includes the communication time between the robot and the PC.

The third most important aspect is the absence of false identifications, and the results from testing are shown in Table 4.

Table 4 Statistical results from the online testing [38]

	Collision not Detected			**Collision Detected**		
	Rule 1	Rule 2	Combined	Rule 1	Rule 2	Combined
Sequences without collision	99.47 %	97.92 %	97.39 %	0.53 %	2.08 %	2.61 %
Sequences with collision	18.86 %	7.55 %	0 %	81.14 %	92.45 %	100 %

Discussion

During the testing phase, all key indicators related to the performance of both mDTW and decision rules were tested. Algorithm implementation aspects were also tested in terms of reliability and reaction time.

The results from both offline and online testing have demonstrated that the proposed mDTW performs well in both offline and in the online application. The mDTW is resilient to different sampling times and non-linear time domain deviations. Therefore, it is suitable for online application in collision detection. However, its biggest advantage may simultaneously be its biggest reason for concern. Testing on a large sample, shown in Table 3 and Table 4, has shown a significant number of missed collisions, when observed independently for each of the decision rules. This fact has demonstrated that mDTW sometimes matches samples so well, that the sample originating from a collision can be mistaken for a nominal operation sample. To prevent this undesirable behaviour, the two implemented decision rules complement each other. Testing on a large sample has shown that zero collisions were classified as nominal behaviour by both decision rules at the same time. This ensures that missed collisions are highly unlikely when the two decision rules are used in parallel.

When results from confusion matrices shown in Table 3 and Table 4 are observed, it is noticeable that there is significant room for improvement. This particularly relates to the refinement of decision rules in terms of their reliability in identifying real collisions. One aspect which has the highest potential to significantly improve performance of existing rules is related to setting thresholds. New rules for setting thresholds should rely not exclusively on the statistical analysis of the signal and but allow for some smart adaptability. One of the ways in which this can be implemented is the moving average and moving variance analysis performed on the sliding window with length determined in accordance with the length of the measurement vector. This implementation would allow higher sensitivity in regions where the dynamics and variance of signals is lower and reduce the chance of false collision in regions with higher activity. For certain applications, shifts in variance calculated in this way could potentially be used also as one of thresholds for detecting collisions. There are also possibilities for designing completely new rules and approaches or extending and improving existing ones. In all cases, complexity of the algorithm should be carefully analysed with respect to processing, and consequently, reaction times.

Collision reaction times depend on algorithm complexity on robot and PC side, the time needed for communication, and time needed for the robot to react to a detected collision. Part of the algorithm implemented on the robot has a simple logic which is possible to implement on various robots. The robot only needs to send a sample of measurements from each of its joints and receive a signal indicating whether a collision has occurred. The communication is performed using Ethernet TCP/IP, and there are limits to the time needed for this type of communication. Other communication protocols can be used to further the collision reaction times.

However, the main concern related to collision reaction time is in the design of DTW and, consequently, also mDTW. With increases in length of the reference signal and/or measurement vector, the time needed to perform matching of signals also increases. The reason for the increase of computing time is directly related to increases in dimensions of the matrix which needs to be formed and searched to perform signal matching. The maximum length of the reference signal depends on various implementation aspects, including sampling times and a hardware platform.

3.3. Detection of intentional interactions

3.3.1. Principal idea

One important aspect of the proposed algorithm with mDTW that has not been addressed in the previous sections is its potential for implementation of intentional interaction detection. Namely, the logic behind the presented algorithm relying on mDTW is completely applicable in the field of detecting intentional interactions. It was mentioned on several occasions that one of arguably the biggest advantages of the non-model-based approaches is the inherent inclusion of the intentional interactions in the algorithm itself. In the presented approaches, the influence of intentional interactions was included in form of their influence on joint currents or torques. It was also mentioned that model-based algorithms typically tend to solve the issue of external interaction forces through integration of a force/torque sensors, although it is not always optimal or adequate.

With the aim of addressing these issues of model-based algorithms, the force-torque sensor in the repetitive applications can effectively be replaced by implementation of the mDTW-based algorithm. From the implementation point of view, there are no theoretical or practical limitations for the presented algorithm to be used in a hybrid mode with a model-based collision detection algorithm, and such possibility will be discussed in this section.

Since collision detection aspect is one of, if not the most important of interaction detection algorithms, different profiles of intentional interaction-induced deviations from nominal values will be compared with deviations caused by collisions in order to discuss the potential and limitations of the proposed hybrid approach.

There are several possible implementation architectures for the hybrid interaction detection algorithm. The principal idea of the algorithm discussed in this section is that the model-based algorithm is in charge of estimating the expected values of the observed joint measurement, which can be either current or torque. The difference between the estimated and measured values, called deviation vector, would serve as the input to the non-model-based part of the algorithm. In the proposed way, all intentional and desired non-modelled behaviour and dynamics of the system would be possible to record using the non-model-based algorithm which would be able to use it as a reference deviation sequence.

Given that the discussed field of application is still related to the tasks which repeat in cycles in identical way, the deviations from the reference values caused by intentional interactions would repeat as well. Consequently, the logic of the mDTW collision detection algorithm presented in the section 3.2 would be applicable also in the field of detecting the intentional interactions.

During normal operation, the part of the algorithm based on mDTW would match the deviation vector of measurement with the reference deviation sequence. The occasion in which samples from the deviation vector cannot be successfully matched with the reference deviation sequence would indicate an unintended interaction or collision and should trigger appropriate response from the algorithm and the entire system.

The following sections are based on [56] and aim to examine the possibility for development of the hybrid approach from the measurement point of view and discuss the implications they may have on the applicability of the solution.

3.3.2. Experiment description

This section intends to examine the effect of load and external forces on the torques of the robot with experiments performed for different types of manipulation and assembly tasks. The unmodelled influence originating from these tasks is compared with the effect of collisions on the internal robot torques. To this end, for each task-related experiment, three cycles of measurements were performed.

First, a torque profile was recorded from a robot executing movement needed to perform the manipulation or assembly task, but without the load or object of assembly itself, so that all recorded torques correspond to signals that would be generated by the optimal robot model which includes only the dynamics of the robot itself.

The second cycle of measurements was recorded from the robot performing the manipulation or assembly task with the load or object which needs to be assembled, therefore also including the external forces related to the load or assembling process.

The third measurement cycle is similar to the first cycle since it was performed on a robot performing its task without the load or assembly object. However, during third measurement cycle, the robot was impacted with different intensities and durations.

With measurements from three cycles, it is possible to better understand the influence of external forces on torques of the robot, and their difference from real impacts. Differences between the second and the first cycle of measurements represent the nature of the deviations the mDTW-based algorithm would receive. Based on the analysis of these deviations, it is possible to understand how applicable they are in the scope of the hybrid approach due to their repeatability and range of values. The difference between the third and the first cycle of measurements is useful for comparison with the deviations from intentional interactions. Their comparative analysis is useful for understanding how applicable and reliable a hybrid approach to interaction detection can be.

Experiments were conducted on Denso VP-6242 6-axis industrial robot with 2 kg of payload. The weight of the object which was manipulated is 1.5 kg, which is 75% of the specified maximum load of the robot. For assembly tasks, the forces needed for the assembly process varied from piece to piece. As mentioned earlier, since VP-6242 does not possess torque sensors, values shown on graphs represent torque values estimated based on current measurements. Measurements range from -100% to 100% of maximum permissible torque for each joint.

3.3.3. Manipulation task analysis

This section analyses the influence of load-related forces that occur during load manipulation tasks. Related, but different tasks were analysed in order to identify possible issues related to unmodelled load.

A movement shown on Figure 27was designed so that it combines pick and place / palletizing and machine tending tasks into one sequence. The robot picks up the load from one of four positions on the pallet, brings it close to the assumed loading position, changes the orientation of the load, and loads it into the machine.

The pick and place / palletizing aspects of the task need to be observed together because of their similarity. The pick and place part is the most interesting from the perspective of the analysis of the repeatability of influence which load picking and releasing has on the torques of the robot when the picking or placing position is the same. The palletizing, or rather de-palletizing aspect of the robot task is interesting to analyse because of different distances requiring different joint configurations.

The first and second pair of picking positions are symmetrical to the robot base which enables examining the effect of load picking to the 2nd and the 3rd axis of the robot since they contribute the most to the vertical movement. For each pair, from the perspective of those two axes, the movement is the same as if the picking was performed from the same point. However, the

distance of second pair of picking points from the robot base is bigger, and it requires different joint configuration. Since the placing point is the same, the effect the load releasing is expected to be the same.

Results shown on Figure 27 confirm that the influence of unmodelled load is very similar for first two picking points. Shape of deviation from the torque measurement without load is repetitive, as well as the intensities of torque differences. Repeated tests from multiple measurements confirm that the load influence on the 2nd axis is predictable and repeatable. Similar conclusion can be made for the second pair of picking points. The repeatability of the deviations from intentional interaction implicates that there is a high potential for application of mDTW-based algorithm for the detection of intentional interactions. However, due to increased distance of the second pair of picking points, the difference in torque caused by the load is slightly higher than for the first pair of picking points as on Figure 27(middle right). The difference in values caused by the different distance from the robot's base can be significant enough to trigger a false collision detection, since the mDTW in the presented form is intended only for identical movements. Releasing of the load caused changes which were also repetitive in intensity and shape resembling a step pulse.

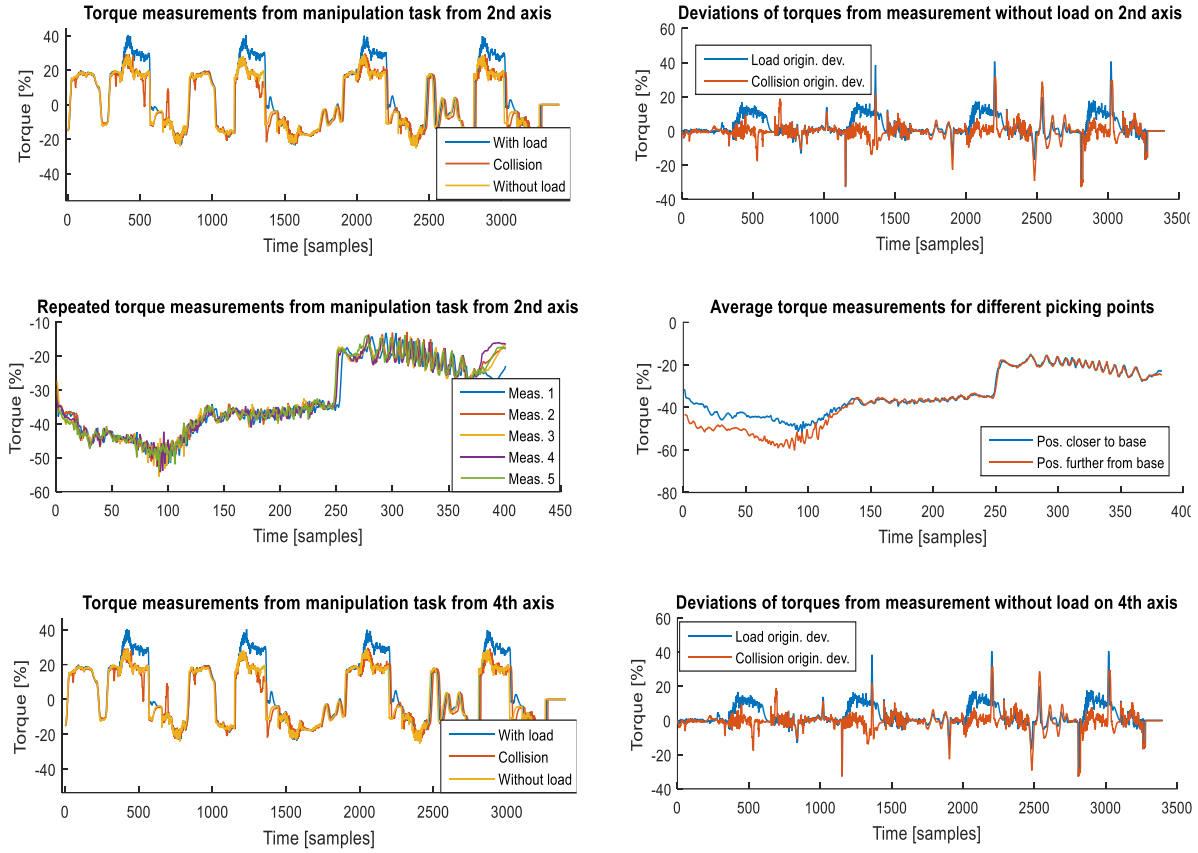

Figure 27. Torque measurements during combined manipulation task [2]. (upper left) Torque measurements with and without load and with collisions. The four segments when the load was picked from different positions are clearly visible as deviations from measurements without load. (upper right) Deviations originating from load and collisions show difference in nature. (middle left) The shown segments of measurements from five consecutive manipulation tasks correspond to periods from when the load was picked from first picking point to the point it was released are repeatable in shape and value. (middle right) Average torque measurements from points closer to the robot base show lower values in sections related to the picking, compared to average torque measurements from those more distant from the base. From the moment the load is close to the loading position, the averaged sections match in values. (lower left) Torque measurements with load are clearly distinguishable from measurements without load from the moment the 4th axis starts to move to change loading orientation. (lower right) Shape deviations of torque originating from the load are repetitive and distinguishable from collision induced deviations.

The machine tending aspect of the robot task is intended to use for analysis of the change of flange orientation to the torque measurements. Unlike the pick and place and palletizing experiments, flange is not oriented vertically downwards in this experiment. Orientation of the flange is influenced by motors in last three axes, which generally have lower power, so the load influence will be more noticeable on them.

Results from Figure 27 (lower left) and Figure 27 (lower right) show that the influence on the 4th axis of the robot is significant, but only from the moment when the orientation of the flange starts to change. With exception to some occurring peaks, the shape of deviation is related to the change of the 4th axis angle, with highest values being achieved at moments when the loading orientation is reached, and slowly decreasing during the final approach to the loading position due to movement of the 5th axis. The conclusions made during the analysis of the 2nd axis are also valid here. Nevertheless, this example also indicates the influence of the change of the joint spatial configuration has on the profile of deviation signal.

In all parts of the manipulation experiment, the influence of load was predictable and repeatable in shape, making it distinguishable from the impacts in most cases when entire signal is observed as it is visible on Figure 27 (upper right) and Figure 27 (lower right). However, the difference between intensities of deviations caused by the load and by collision is not very big, and occasional peaks make them more difficult to tell apart, especially if the distinction needs to be made in matter of samples.

3.3.4. Assembly task analysis

Assembly tasks often require the robot to exert some force in order to join two parts. The following examples analyse the influence of interaction forces during three different assembly types.

Snap fit. Snap fit is a very common type of assembly process performed by robots. Variations of this process can be found in different fields of industry, depending on the assembly requirements. The experiment was conducted on pieces with annular snap fit, but derived conclusions are applicable on any type of snap assembly.

The results from the testing are shown on Figure 28. The two peaks on the image originate from two levels of latches and corresponding grooves. It is also noticeable that the second peak has higher value, since the force needed to deform both levels of latches is higher than the force needed to deform only the first level of latches. When the difference between signals recorded with and without assembly object is compared, it is possible to notice that it resembles a peak belonging to an impact. Repeated tests have shown that while the general shape of the torque is constant, slight variations in the position of the assembly object affect the intensity of peaks. The profile of deviations makes them unsuitable for the mDTW-based detection algorithm because the intensity and shape make them difficult to distinguish, especially in short time. However, repetitiveness of the tasks for which the proposed hybrid approach is intended enables additional information related to periods in which the intentional interaction is expected. Therefore, a possible solution might be to interpret all deviations outside this time period as unintentional contacts, i.e., collisions.

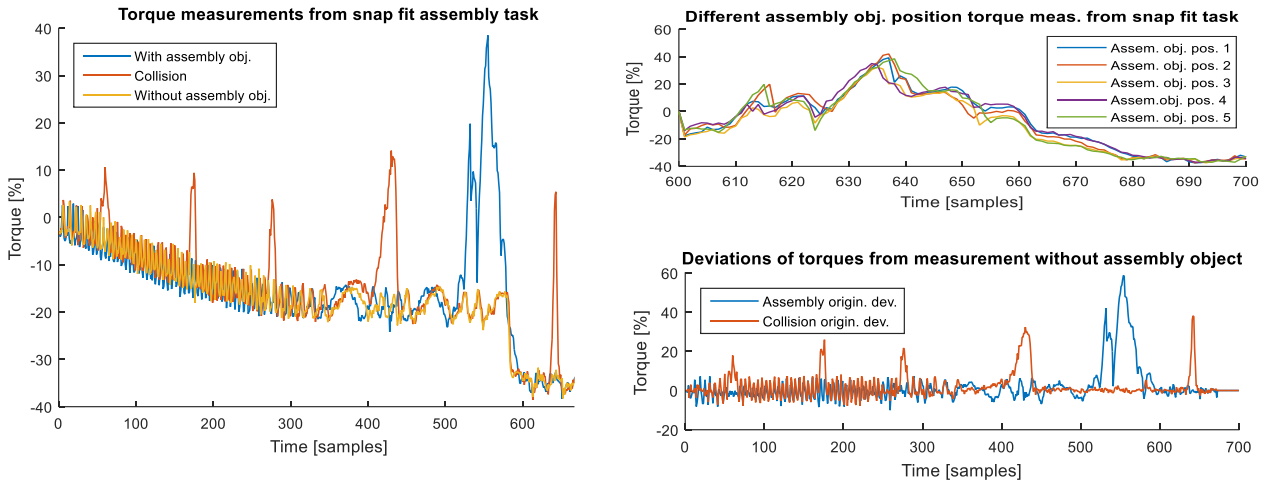

Figure 28. Measurements from snap fit assembly [2]: (left) Torque measurements of robot operation with and without object of assembly as well as with collisions. The double peak originated from two levels of latches and grooves of the object of assembly. (upper right) Five consecutive measurements of assembly task show similar shape, but also differences in intensities in some areas which are result of slight variations of position of the assembly object. (lower right) The shape of the deviation originating from the assembly is similar both in shape and in intensity to the shape of deviations originating from collisions and requires special attention.

Spring latch. This type of joining parts is present in applications where it is required to have a good contact, but also allow for easy disassembly. In order to connect the pieces, it is first needed to force the inserted part against a spring, and then move it to a position in which the compressed spring will hold it in place. Assembly and disassembly were performed in a single movement with a two second break between them.

Results on Figure 29 have shown significant influences on torque, most notably on the 5th axis of the robot. Measurements signals with and without actual assembly piece show great difference, both in assembly and disassembly movement. Although shapes of deviation are roughly symmetrical, tests have shown some differences in intensities of torques exerted on the robot during assembly and disassembly. While collisions with similar shape as parts of assembly induced deviations have been recorded during testing, the overall shape makes them distinguishable from collisions. In particular experiment, the spring induced high torque differences which greatly surpass those of impacts. However, in general case, intensities of assembly and collision induced deviations may happen to be very similar, making them more difficult to tell apart, implying that reliability of mDTW algorithm application would need to be analysed on a specific case-based analysis. Similar to the conclusion for the snap-fit assembly, addition of the period in which the intentional interaction is expected would increase the reliability and applicability of the proposed approach.

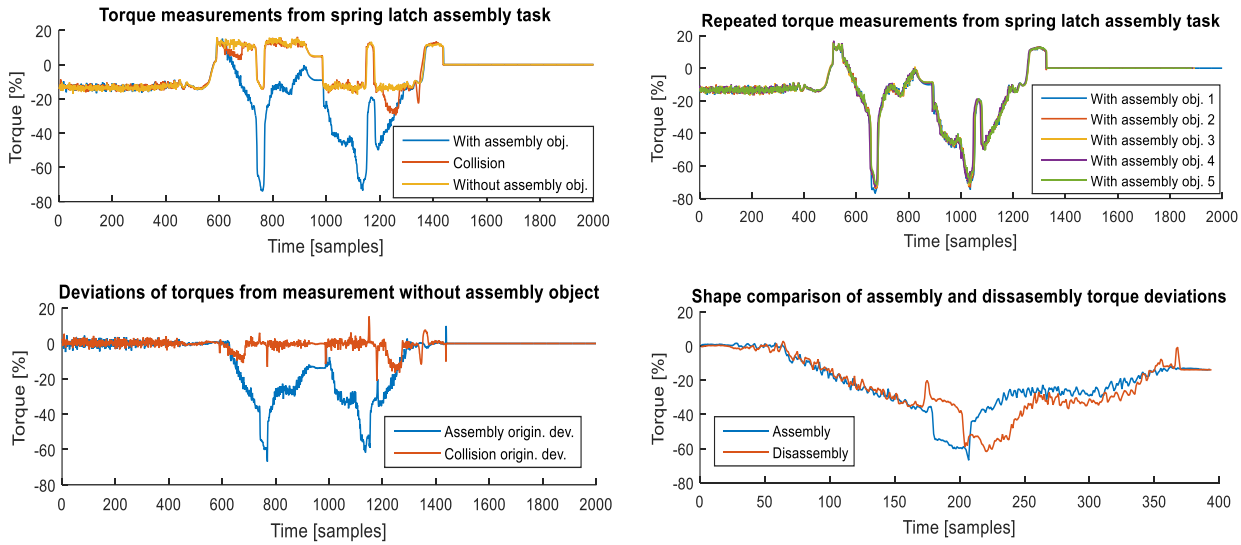

Figure 29. Spring latch task results: [2] (upper left) Torques during assembly and disassembly movement. The deviations originating from the assembly task greatly surpass those induced by collision. (upper right) Torque measurements from 5 consecutive recordings show great similarity and repeatability. (lower left) Deviations induced by assembly task and collisions. Assembly originating deviations show elements of symmetry, but with slightly different intensities. Some collision deviations match sections of the deviations originating from assembly. (lower right) Assembly torque deviation and mirrored disassembly deviation show elements of symmetry.

Screwing. Although commonly performed by screwing devices mounted on a robot as a dedicated tool, many robots with unlimited rotation on the 6th axis are used in applications where a part itself is screwed onto another part of the assembly, rather than two parts being fastened by a screw. This is especially the case when a part needs to be rotated less than entire circle in order to screw it in place. In this experiment, both screwing and unscrewing operations were performed and analysed.

According to expectations, experimental results have shown that the biggest torque difference is on the 6th axis of the robot. The shape of deviation shown on Figure 30 resembles a ramp signal profile, which corresponds to the nature of the process. In the particular case, the difference can be distinguished from the real collisions due to its duration and intensity, as well as its shape. However, it is possible to imagine that some screwing assemblies might require a higher force. More importantly, collisions need to be identified as soon as possible, which means that decision cannot wait for several samples to make the distinction. The unscrewing action resulted in decrease in intensity of the torque originating from the unscrewing, as it can be seen on Figure 30 (right). However, the shape of deviation from the measurement without the assembly object has maintained the same shape and nature as with the screwing process. Results from the analysis of this type of assembly task imply that it would be possible to effectively implement the proposed hybrid detection approach, since the mDTW algorithm would be able to distinguish between the intentional and unintentional interactions, as long as there is an adequate difference in intensities.

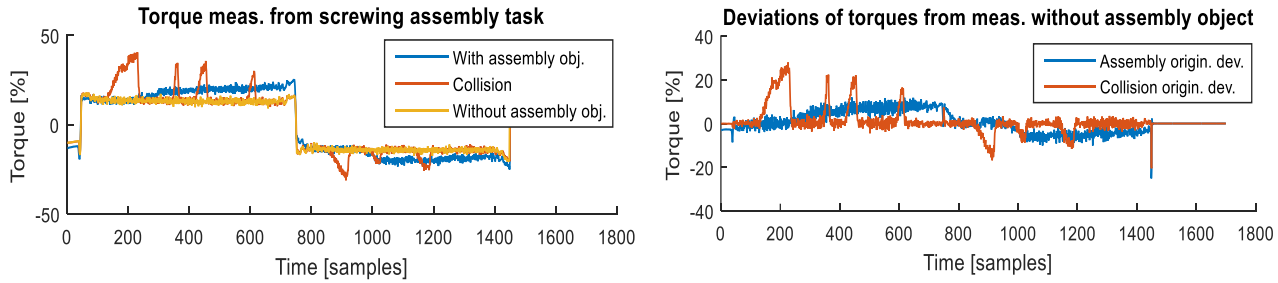

Figure 30. Screwing assembly task [2]: (left) Torque measurements from screwing and unscrewing action resemble ramp signals due to changing forces that act upon the 6th axis. (right) Deviations of torques originating from the task execution and collisions. Though sometimes similar in intensity, the shape of deviations is clearly distinguishable in terms of shape.

3.3.5. Discussion of results

The aim of this analysis was to examine the influence of unmodelled intentional external forces on the quality of collision detection and their potential incorporation into a hybrid detection approach with possibility to interpret intentional interactions. To that end, two types of applications which regularly incorporate forces of different profiles and nature were chosen as typical examples.

Experiment with manipulation tasks was performed in a single experiment combining pick and place/palletizing and machine tending tasks. Experiments have shown that the influence of the unmodelled load-related forces has varying effect on the torque measurements depending mainly on the position and orientation of robot flange and consequently the joints affected. This observation is irrelevant for the application in strictly repetitive tasks, but it is one of the most important aspects which will be addressed in the 4th Chapter in which the identification while performing similar movements is considered. It was shown that the disturbances in measurements with and without load can easily be mistaken with collisions with regards to their intensity. However, with exception to variations in intensity, it was shown that the shape of disturbances is repetitive, and that it does not depend on the configuration of robot segments. The aforementioned is important because it means that a disturbance of certain shape can potentially be excluded from being identified as collisions. In [12] it was shown that intentional contact can be identified through use of low and high pass filters with corresponding rules, which implies that influence of load-related disturbances can potentially also be classified as intentional using similar principles with some modifications. The main risk lays in the fact that collisions and unintentional disturbances need to be identified as soon as possible, making it harder to enable processing time needed for avoiding misinterpretation of intentional contacts. However, the processing and interpretation delay are much lower with the approach in which mDTW is used to match deviation vector to the profile of the reference deviation sequence.

Three types of assembly examples were used for experiments. Snap fit assembly example has shown that profile of torque disturbances it generates has a high potential to be classified as a collision. In particular test case, both the intensity and duration of the peaks in torque originating from connecting two parts closely resemble those of a real collision. Furthermore, it was shown that slight deviations of position of the assembly parts cause variations in intensity of force required to join them. A potential solution to the problem of false identification can be in having higher tolerances for collision detection in a limited time interval within the assembly cycle. Unfortunately, such tolerance modification or introduction of the time interval for intentional contact could lead to higher risks of faulty implementation, and potential missed detection of real collision, so this issue deserves special attention.

Forces originating from spring latch assembly example have a high potential to be misclassified, primarily due to their intensity. While shapes of disturbances induced by this assembly type have repeatable profile in particular test, in general case it may depend on positioning accuracy of the assembly parts and repeatability of the spring stiffness. As a general conclusion on this example, the repeatability of the assembly process is the main factor influencing whether forces in this type of assembly can reliably be interpreted as intentional, without risking missing real collisions.

Assembly by screwing experiment has shown that the deviations from the measurements from the robot without actual load are repetitive both in shape and in intensity. Although the intensity of the torque deviation was not very significant in the performed experiment, in general case the torques needed to apply can easily rise to much higher levels and be misclassified as collisions. Nevertheless, with adequate positioning accuracy of the assembly parts, a solution can be found to interpret forces during this type of assembly as intentional, in the proposed hybrid approach architecture, or even as in [12].

As a general conclusion with regards to effect that unmodelled dynamics has on collision detection, the experimental results have shown that influence of unmodelled intentional forces should not be neglected. Failing to do so may lead to false collision detections, while simple

increase in detection thresholds may even lead to human injuries. It was shown that some types of forces are predictable and repeatable in shape, making them easier to model and/or include into collision detection algorithm in some other manner. Unfortunately, there are forces which are very difficult to distinguish from the collision because of their shape and/or intensity.

Prompt and timely collision detection poses an additional challenge even for forces which have been shown to be distinguishable from collisions as their correct classification may require additional time for processing.

In a broader sense, varying and unmodelled external force influences have been examined and dealt with in context of impedance control [57] [58], which can be one of directions to elaborate on this topic in future.

Nevertheless, the analysis of deviation has shown that algorithms based on mDTW have significant potential also for detection of intentional interactions when they are used to observe deviations from the expected joint measurement values. To this end, their hybrid implementation with model-based algorithms would make them a viable and potentially preferable alternative to implementation of force/torque sensors. The biggest advantage over integration of force/torque sensors at the tool flange is that they can detect collisions on all joints from the base of the robot to the robot's flange, and not only interactions that the tool has with the surroundings.

3.4. Chapter discussion

The intention of the presented approaches was to offer a collision detection algorithm as a contribution towards safer workspace sharing and future collaboration between humans and industrial robots without external sensors or open control architecture. Compared to most existing methods, the previously presented collision detection algorithms are not based on a robot model. Instead, they relay on the fact that vast majority of robot tasks are repetitive and uses measurements from one cycle of task execution as reference values for all succeeding operation cycles. While this concept offers lower flexibility compared to model-based approaches, it benefits from reduced complexity and inherent inclusion of various aspects of the robot motion which are difficult to model. These may include unknown weight and dynamic properties of part in manipulation tasks, intentional contact forces during assembly, variable weight in dispensing applications, and many other aspects which are either complex or time consuming to model.

The first realization of the algorithm for collision detection was intended for the integration on the robot controller itself, as an attempt at reducing reaction time by avoiding time needed to exchange data with another device. To overcome issues related to the uneven sampling, two decision rules were set in place to cover regions with different signal dynamics. However, the way in which the detection thresholds are determined is suboptimal, as it requires understanding the reasoning behind the detection rules and some knowledge of signal analysis. The threshold setting can be translated into an automatic procedure, but it would require larger recorded sequence sample, which would affect the algorithm's applicability on older versions of robots. For better reliability, the algorithm was designed with inherent safety precautions. It works in background of all tasks created by user and starts automatically by switching to mode for automatic task execution. It can be protected from unauthorized access, and it stops the movement of the robot if the collision is detected, regardless of the other programs. In order to run properly, i.e., to minimize number of false collision identifications, it requires a triggering signal from the main robot program. However, if this signal is omitted, the algorithm will not allow any motion of the robot, preventing injury or damage.

Although it performed quite well, the algorithm implemented on the robot controller served best as an introduction to the collision detection topic, and a solid base for designing a version intended to be implemented on PC.

The attention to designing an efficient and performant collision detection algorithm was shifted to implementation on a PC permanently connected to the robot controller using standard communication protocols. The reason for this was the increased complexity, which meant it could no longer be fully implemented on controllers of robots with closed control architecture. However, permanent connection to a PC or similar processing unit should not be an issue, having in mind increased reliability and sensitivity.

As a corner stone for reliable collision detection, this approach presented a modification of Dynamic Time Warping method (mDTW) which was used in order to overcome sampling uncertainties imposed by the robot controller design. Compared to traditional DTW [51], it offers two advantages of key importance for Collision detection application. Most importantly, unlike [51] [53] [52], it allows matching signals with different content. That means that one signal can be compared to a part of another signal with similar content or shape. This modification also allows the online application of DTW, making it useable for the presented collision detection concept. At the same time, the mDTW preserved the ability of DTW to optimally match signals by compressing or dilating the time axis. Unlike some other proposed modifications of DTW [55], the mDTW does not allow skipping samples. This ensured that the samples indicating a collision, or some other anomaly cannot be neglected. The reduced length of compared signals enables shorter processing time, without early pruning or other techniques proposed for speeding up DTW [54]. Unlike EDIT distance [48], Hirschberg's algorithm [49] and Approximate string matching [48], the proposed method is applicable in applications with infinite number of signal values, and it does not allow skipping samples. Compared to other signal matching techniques, such as Partial Curve Mapping [50]and Discrete Fréchet Distance [50], the mDTW is intended for application with time series.

Additional contributions are the two complementary decision rules. They rely on mDTW matching ability to detect deviations from reference signal in terms of absolute difference and Eigen values. While the idea to observe absolute differences between samples is not new, mDTW matching allowed lowering detection thresholds. This increased the sensitivity of the algorithm and allowed detection of lower intensity collisions. The rule based on Eigen values observed the distribution of matched sample pairs, offering a new perspective and additional safety measure.

The proposed algorithm was intended for realistic application on various brands and generations of industrial robots with closed control architecture. Therefore, signals which are most commonly available from all robots, such as current and torque estimate, were used for the purpose of detection. Throughout the Chapter, various aspects of the algorithm were considered and designed with implementation in mind. The procedures for collision detection were designed to be the least intrusive to the normal programming of the robot and to provide maximum safety. All procedures, both on robot and on PC are fully automatic, and they require very low setup time. The only intervention required from the user at setup, and/or after changing of the robot main movement task, is to set correct IP address of the robot, and to initiate the procedure on PC.

In accordance with its intended purpose, the algorithm was also experimentally validated on Denso VP 6242 and ABB IRB 120 6-axis industrial robots, and results were discussed from reliability and implementability point of view. Conclusive results show that collisions are efficiently detected even in the early stages of the algorithm development, and that safety within the shared workspace is significantly improved.

Nonetheless, improvements on the algorithm itself can be made in reduction of reaction time by improving existing or using more efficient calculation strategies for signal matching and focus on improving the sensitivity of the algorithm even further. Examining the possibilities of incremental computing, such as the one presented in [52], and its adaptation to the specifics of the mDTW has a potential for great improvements in computational requirements and reaction speed. One important field for improvement includes improving the flexibility of the algorithm and its capability to adapt to a family of similar movements without having to re-set reference values. To this end, solutions will be considered in the next Chapter.

The consideration related to the implementation of the mDTW-based collision detection in the field of detection of intentional interactions has shown its potential for application in that field

as well. Such implementation does require observation of the deviation from expected values, which led to the conclusion that a hybrid approach, in which deviations would be generated based on the experimented values of the model-based algorithm. Suggested approach would solve many of the problems related to model-based algorithms and contact tasks or unmodelled dynamics. Additionally, it would extend the field of application of mDTW as a non-model-based algorithm.

The analysis of the representative contact tasks has shown that there are applications in which such hybrid approach can be very successful, due to the differences of collision induced deviations and those originating from the intentional interactions. More importantly, it has pointed out tasks in which such distinctions cannot be made reliably, especially given the requirements for short reaction times.

The downside of use of the mDTW-based intentional interactions detection are related to the fact that it is limited to the identical repetition of the task, which narrows the field of application for such approach. The possibilities for extending the application field of the mDTW-based algorithms to movements which require different spatial relation of robot joints than those during the representative referent movement are examined in the next Chapter.

4. DETECTION WHILE PERFORMING SIMILAR TASKS

This Chapter aims to present strategies and solutions related to identification of interactions while performing tasks which do not repeat in identical cycles but perform the same type of task [59]. Compared to the movement or operation made during recording the reference sequence, these tasks include movements and operations performed under different spatial relation of robot joints, i.e., different joint configurations. This augmentation aims to offer a completely new quality to the previously presented algorithms for collision detection, allowing them to be used under different conditions to those when the reference sequence was formed. From the perspective of load manipulation, rather than pick and place operations from and onto same positions, these tasks may include palletizing or depalletizing operations, bin picking, camera-guided pick and place operations, machine tending etc. From the processing tasks, examples may include riveting, drilling, stamping, snap-fit assembly, bending and other operations which can be performed with different orientations of the tool and joint configurations.

The aforementioned greatly extends the potential field of applications, but it requires additional analysis and information from the robot controller. This information is primarily related to understanding the joint spatial distribution and its influence on individual joints.

Additional and highly valuable benefit of the alteration proposed in this Chapter is related to its extended possibility for implementation in field of detection of intentional interactions, in a standalone or hybrid capacity. Therefore, the analysis in the following sections will be mostly considered from the perspective of a hybrid approach in which only the deviations of the measured signal from its nominal values is observed, as explained in the section 3.3.

4.1. Background idea and applications

Considerations from the introductory part of this Chapter and conclusions related to the previously presented mDTW approach form the basis on which the approach and algorithm presented in this Chapter are conceived. The approach will be designed to be implementable without expert knowledge, to work without additional sensors, and to be platform independent. To achieve these goals, it relies on previously developed algorithms for the identification of kinematic parameters and, more importantly, for non-model-based collision detection. The most important contribution presented in this Chapter will be in the augmentation of the modified Dynamic Time Warping (mDTW) algorithm in the form of the inclusion of coefficients related to a kinematic model of the robot. In this way, an entirely new dimension to the algorithm is added, enabling it to respond to changes in spatial relation of robot joints and the effects that external contact forces have on them.

All research and procedures relevant to the realization and implementation of the proposed algorithm will be briefly overviewed to understand better the information required and the level of automation it is possible to achieve, both of which are important aspects from the Industry 4.0 perspective. However, the focus will be on the fusion of this information to enable an effective and reliable algorithm for the inclusion of contact task dynamics. To simplify the analysis, an industrial robot with six revolute joints is considered, being the most common configuration, but all conclusions are also valid for other robot configurations as well as robots with prismatic joints.

The starting assumption is that when a robot is performing a task that does not involve contacts or changes in weight, values of all measurements that are available at each robot joint are in accordance with nominal values, i.e., can be considered to be known based on a nominal model of the robot or in some other shape or form. Contacts with the surroundings or changes in weight cause deviations from these nominal values. The idea underlying the proposed approach and algorithm is that these deviations reflect contact task dynamics and that their correct interpretation can lead to the implicit inclusion of the unmodelled forces and torques that appear during contact tasks.

In this Chapter, deviations recorded while performing one representative example of the contact task are referred to as a reference sequence. Deviations recorded during the operation of the robot stored in a sequence with first-in-first-out logic will be referred to as the measurement vector. The correct interpretation of the deviations in the measurement vector based on its comparison with the reference sequence can indicate whether the deviation originates from an expected, desired contact whose dynamics was not included in the nominal model or from some erroneous state or condition.

As presented and discussed earlier, the mDTW enables matching the measurement vector to a segment of the reference sequence as well as implementation in real time [53]. It also has the advantage of implicitly including all dynamic events and phenomena without modelling or deep understanding, since it observes joint events, which are consequences of task dynamics. However, the drawback of mDTW is that it was designed to perform matching of signals while the robot performs the movement in an identical or very similar way, maintaining a similar joint posture. Periodic tasks that include such movement are very common, but in the general case and especially in Industry 4.0 applications, a robot is expected to be more agile and adaptable to meet the needs of the interconnected event-driven production environment. These requirements also relate to its movement, meaning that the straightforward implementation of mDTW is no longer suitable in the general case, since the movements for the same type of task may require significant changes in the spatial relation of the robot's joints. Differences in joint spatial relation dictate differences in effects that contact forces have on individual joints, resulting in the different profiles of deviations that are noticeable on a joint level. Supplementing mDTW with values related to the kinematic model of the robot aims to solve these drawbacks, and to this end, the following sections describe procedures used to identify kinematic model of the robot and its end-effector as well as the way in which these parameters are introduced into mDTW.

4.2. Identification of robot DH parameters

Starting from identifying robot Denavit-Hartemberg (DH) parameters from the base of the robot to the flange to which the end effector can be attached, it is essential to note that it can be performed in numerous ways, including manual calculations. However, the general approach of algorithms in this thesis is to facilitate the implementation of the algorithm and reduce the chances of errors wherever possible using automated procedures. To this end, this section aims to present an approach to designing an automatic procedure for obtaining DH parameters of a robot with arbitrary configuration. The approach is based on obtaining partial pose measurements, containing only the positions of the point of interest, and using this information to perform a full body identification of kinematic parameters.

Previous research [60], [61] [62] has described that an automatic procedure that moves the individual axis while monitoring the position of the point of interest at the end effector can be used to perform the identification of kinematic DH parameters of the robot, or some of their alternatives [63]. The procedure can effectively extract relevant parameters for rotational and prismatic joints alike, and therefore the configuration of the robot does not impose any applicability limitation. A similar principle was later further elaborated [64] [65] with considerations regarding accuracy and tools that can be used for its successful implementation in realistic environments.

Common to all the mentioned identification approaches is that the DH parameters are calculated based on the trajectories that the observed point of interest has while the robot moves each of its axes individually. In the case of rotational joints, the trajectory will be circular, and the centre and orientation of the path will indicate the direction of the z axis of the observed current joint. From the formed spatial directions of z axes for each joint, directions of other axes can be calculated as well. In the case of non-parallel z joint axes, the common normal from the previous towards the current direction of the z axis will determine the direction of the new x axis, and the point of its intersection with the current z axis will determine the current coordinate origin.

The approach to the identification presented in the following sections is based on [60] and it introduces a different way of calculating parameters using only three measurement points per joint. This approach enables complete definition of joint rotation axis based on analysis of two measurement vectors and their bisectors, unlike method presented in [62] which uses fitting measurements to the arc in order to determine the rotation centre. Furthermore, unlike in [62] the directions of *x* and *y* axes of each joint are determined in accordance with DH convention for facilitated further calculation of kinematic parameters. For better understanding, it additionally differs from the [61] by its representation of vectors using direction and point, rather than Plücker coordinates.

The benefit of using the aforementioned approach is that it is not restrictive regarding the type of sensors that are used to determine the position of the observed point in some external reference frame. The position can be measured directly as long as the accuracy and measurement volume are not issues.

A theodolite or any other measurement sensor can be used. Another approach is to measure the position indirectly, using joint encoder measurements and a built-in kinematics model all robots use for direct kinematic tasks. For the purpose of performing the analysis in following sections, the kinematic model was calculated using the coordinates of the TCP in the robot's base frame, which were provided by the robot's controller based on the measurements of the joint encoders.

4.2.1. Denavit-Hartemberg notation

Starting from the notation on which the analysis and development of the approach is based, this section offers a quick overview of the terms and notations used in the following discussion and elaboration, illustrated on Figure 31.

In order to describe a relation between two joints with indexes *i* and *i-1*, i.e., coordinate systems related to them, using D-H notation, two preconditions must be met [66]:

- axis x_i is perpendicular to axis z_{i-1}
- axis x_i intersects axis z_{i-1}.

For the base coordinate system, the *z* axis goes along rotation axis of the joint. The *x* axis can be chosen in any suitable direction, as long as it is perpendicular to the *z* axis, and *y* axis is set in such way that it forms a right-handed Cartesian coordinate system together with previously set *x* and *z* axes.

For a joint with index *i*, z_i axis is also set along its rotation axis. However, the x_i axis is chosen in such way that it is positioned along the vector perpendicular to both z_i axis and z_{i-1} axis of previous joint, which is why it is also known as common normal.

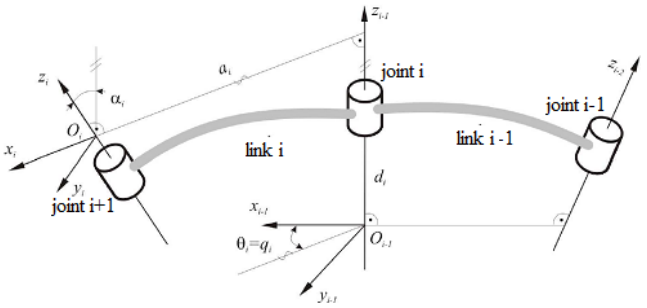

Figure 31. Notation used for calculation of Denavit-Hartenberg parameters [66].

To Denavit and Hartenberg, the common normal served as the main geometrical concept which enabled them to find a minimal representation [67]. This normal also represents the shortest distance between axes z_i and z_{i-1} The origin O_i of coordinate system is located at the intersection of

z_i axis and the previously determined axis x_i. The y_i axis completes the right-handed Cartesian coordinate system.

In order to match coordinate systems of two neighbouring joints with indexes *i* and *i-1*, a set of two translations and two rotations was used. First, the coordinate system with index *i-1* is translated along axis z_{i-1} to the point where it intersects with axis x_i. The distance of translation represents parameter d_i. Second operation rotates the coordinate system with index *i-1* until axis x_{i-1} is aligned with axis x_i. The angle of rotation is equal to parameter α_i. The third step is to move the coordinate system with index *i-1* along axis x_i until the origins O_i and O_{i-1} match. Distance travelled along x_i axis is equal to parameter a_i. The final step is to rotate coordinate system with index *i-1* around axis x_i until axes z_i and z_{i-1} match. The angle of rotation represents parameter θi.

All four steps of matching two coordinate systems can be described with set of four acquired parameters θ_i, α_i, d_i, and a_i, and a homogenous transformation matrix (14) - (15) [66]:

$$H_{i-1}^{i} = Rot_{z,\theta_i} \cdot Trans_{z,d_i} Trans_{x,a_i} Rot_{x,\alpha_i} \tag{14}$$

$$H_{i-1}^{i} = \begin{bmatrix} C\theta_i & -S\theta_i C\alpha_i & S\theta_i C\alpha_i & a_i C\theta_i \\ S\theta_i & C\theta_i C\alpha_i & -C\theta_i S\alpha_i & a_i S\theta \\ 0 & S\alpha_i & C\alpha_i & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{15}$$

4.2.2. Obtaining Parameters

This section describes an algorithm that can be used in order to identify Denavit-Hartenberg (DH) parameters. The concept is based on gathering partial pose measurements of a single point attached onto robot's end effector. During the acquisition of measurements, the robot performs elementary movements, and therefore it does not require any complex programming. Additionally, this approach can be fully automatic. Although similar movement procedure is used by company Scape Technologies to extract DH parameters from robot itself, the calculations are based on a different concept and for different purposes. The following analysis is considered for a 6-axis robot with revolute joints, but it is applicable to any other configuration as well.

The idea is to gather exactly the information which is needed to calculate DH parameters, and that is the relative position and rotation angles between neighbouring axes.

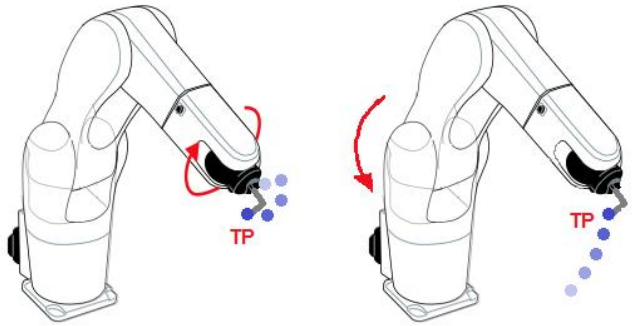

Figure 32. Examples of robot movements and measured positions [60].

In order to perform measurement, it is needed to measure position of a point rigidly fixed to the last segment of the robot. Let us name that point of interest as tracked point, or TP. The only restriction to the position of TP is that it may not be located on the rotation axis of the last joint. The restriction is imposed by the principle of the algorithm itself.

If the TP is not collinear with the axis of the rotation of the last joint, when rotation of that particular joint occurs, the TP will have a circular trajectory, as shown on Figure 32. The rotational

axis of the observed joint is parallel to the vector which is normal to the plane in which the measurements are located and runs through the centre of the measurement circle.

It is possible to identify the needed values in numerous ways. However, having in mind the restrictions that are often imposed by various factors including the construction of the robot, measuring volume or accuracy of the measuring devices, robot surroundings etc., it is desirable to keep in mind the applicability and implementation aspects of the solution. In order to define the two defining aspects for the rotational axis, i.e., the direction and the point at the centre of the measurement circle, an approach described in this section relies on the possession of measurements in only three points per observed joint.

Let us assume that the three measurement points A, B and C located on the circular trajectory of the TP after movement of a single joint were acquired as the first, the second and the third measurement. In that case, vector product of vectors AB and BC would result in the vector parallel to the joint rotation axis.

After normalization of this vector, the next step in defining the rotation axis is to specify the point through which the rotation axis runs, i.e., the centre of the circle. The centre of the circle can be defined as the point of intersection of the vectors AB_n and BC_n, bisecting the vectors AB and BC respectively in the plane defined by the three non-collinear points A, B and C, as shown on Figure 33.

Although theoretically inevitable, due to the nature of the measurements, it is expected that the bisecting vectors will not exactly intersect with each other, so the point of interest is actually the point in which they are the closest. Having acquired the point of interest, it is possible to define the exact joint rotation axis, which is by the definition of DH notation the *z* axis of the observed joint.

The procedure described for the last joint can be repeated for all joints, and once the directions of all robot joints, i.e., all of their *z* axes have been calculated, it is possible to extract the DH kinematic parameters of the entire robot.

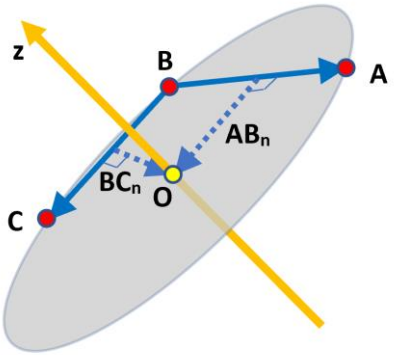

Figure 33. Calculation of the joint rotation axis.

When points and direction vectors of *z* axis of two neighbouring joints *i* and *i-1* are observed, the shortest distance can be calculated as a vector connecting two points on axes, while being perpendicular to both axes. This vector is also known as a common normal, and its length represents parameter a_i. Point of intersection of this vector with axis z_i determines the origin of the coordinate system of joint *i*, and its direction determines the direction of axis x_i, which points away from the previous joint.

In the case of parallel z_i and z_{i-1} axes, the direction of the x_i axis is determined using identical rules. However, since the number of the common normal is infinite, any convenient point can be chosen as the current joint coordinate system origin. The y_i coordinate axis is determined to complete the right-handed Cartesian frame, forming the joint coordinate frame. The spatial relation of all joint coordinate frames can then be used conventionally to determine kinematic parameters in DH or any other notation.

The distance between the coordinate origin O_{i-1} of joint *i-1* and point on z_{i-1} closest to the axis z_i represents offset d_i.

When coordinate origins have been matched, axes x_i and x_{i-1} lay in the same plane. Therefore, the angle θ can easily be calculated from scalar product of unit vectors i_i and i_{i-1} along x_i and x_{i-1} (16), respectively.

$$\theta_i = arccos(i_i \cdot i_{i-1}) \tag{16}$$

After rotation of axis x_i to match x_{i-1}, the angle between z_i and z_{i-1} can also be calculated using scalar product of unit vectors k_i and k_{i-1} along respective axes. Calculated angle represents angle α_i (17).

$$\alpha_i = arccos(k_i \cdot k_{i-1}) \tag{17}$$

Calculated values form DH parameters for one set of joints, which can be incorporated into homogenous transformation matrix (15). When values for all neighbouring joints have been determined, the final transformation matrix is equal to product of all matrices (18).

$$H_1^n = H_{final} = \prod_{i=1}^{n} H_{i-1}^i \tag{18}$$

When final transformation matrix has been obtained, the model can be used to accurately represent the real robot. Parameters α_i , d_i and a_i are constant in case of rotary joints, while θ_i are actually internal coordinates q_i, used to calculate the position of segments. While the described procedure has been explained on example for robots with rotary joints, it is also applicable for robots with linear axis with simple modifications.

If the robot has linear joints, there are a few differences, some of which simplify calculation. One difference is that axis z_i is set along the axis in which the linear joint moves. Value a_i is considered to be zero since it can be chosen arbitrarily. Axis x_i is set to be normal to the plane in which z_i and z_{i-1} lay, i.e., to be in direction of $z_{i-1} \times z_i$, or the opposite direction. Axis y_i is set so that it forms a right-handed Cartesian coordinate system with x_i and z_i. Value d_i is now internal coordinate q_i, and it is equal to zero at the point where O_i and O_{i-1} match. Parameters θ_i and α_i are constant in case of a linear joint.

From the described procedure, it is possible to conclude that the approach can be applied for any number and type of joints with single degree of freedom. Therefore, it can be used with any given configuration of the robot, including external axes that may be used to extend its robot's working range or to introduce redundancy, as long as they form a kinematic chain with robot itself.

4.2.3. Results and Discussion

Based on the previously described algorithm, parameters of Denso VP-6242 robot were calculated. The measurements obtained from the robot controller in form of full-pose measurements were calculated on the controller itself using its kinematics model using measurements from joint positions/angles. This model used by the robot controller for forward kinematics calculations, and it is commonly found on almost all industrial robots in use. Full-pose measurements include the positions and orientations of the point in robot's base coordinate system, but for calculations, only the positions were used, while rotations about each of the axes were omitted during calculations. The results of calculations is presented in Table 5.

This approach to calculation of DH parameters for the use within the KA-mDTW algorithm is very convenient, as it does not require any external sensors or equipment. Additional advantage of the algorithm presented in this section is that it can accommodate for constrictions imposed by

the physical design of the robot or its working environment. The former was most useful for the application on this robot, as its 3rd axis does not have the full range of motion, as mentioned earlier. For that reason, the acquisition for all joints was performed in the configuration in which the 3rd joint was at 90°. The consequence of this configuration are visible in Table 5, where Θ for the 3rd joint is 90°. This however, is easily compensated by the algorithm by means of simple subtraction performed when forward kinematics calculations are performed.

Table 5 Calculated spatial axis distribution and joint DH parameters

	Calculated axis directions and relevant points				**Calculated DH parameters**			
	z axis	*Cicrle centre (x,y,z)*	*x axis*	*Origin (x,y,z)*	*a [mm]*	*d [mm]*	*α [°]*	*Θ [°]*
1st joint	(0, 0, 1)	(0, 0, 515)	(1, 0, 0)	(0, 0, 0)	0	280	90	180
2nd joint	(0, 1, 0)	(0, 0, 280)	(-1 ,0, 0)	(0, 0, 280)	210	0	0	90
3rd joint	(0, 1, 0)	(0, 0, 490)	(0, 0, 1)	(0, 0, 490)	75	0	-90	0
4th joint	(1, 0, 0)	(380, 0, 565)	(0, 0, 1)	(0, 0, 565)	0	210	90	0
5th joint	(0, 1, 0)	(210, 0, 565)	(0, 0, 1)	(210, 0, 565)	0	0	90	180
6th joint	(1, 0, 0)	(380, 0, 565)	(0, 0, -1)	(210, 0, 565)	0	70	0	0

From the implementation point of view, the procedure is very simple, and it can be divided into two main phases. The first phase is dedicated to the acquisition of measurements. The robot needs to move only one joint for each set of measurements. Therefore, its program only needs to move the arm in joint coordinates in certain angular increments. Angular increments and number of samples in general depend on the physical capabilities of the robot and computational requirements of the algorithm. Although only three non-collinear points are required to perform the algorithm in the proposed way, it is recommended that they are chosen properly in order to cover widest range of movement of each joint with adequate resolution. Robot - intended program is repeatable for each joint, so one function can be reused, cutting down on programming time, and making it easier to adapt for various brands of robots.

The second phase of model acquisition is the analysis of measurements and calculation of DH parameters. Homogenous transformation greatly simplifies computation requirements, as it reduces number of operations. Accurate acquisition of measurement points is one of most important aspects of the algorithm since its quality directly or indirectly influences the accuracy of many other values. Additionally, more samples generally result in better outcome. However, simply increasing the number of samples while measuring them in limited range of joint's movement cannot bring optimal results by itself. As mentioned before, measurements should be taken from entire range of motion of one joint, in order to get more robust calculation.

It is possible to note that the described procedure only requires partial pose measurements, i.e., positions of points in space. Orientation of points is not necessary for any of the calculations, as all the needed information can be extracted from position of points.

4.3. Identification of end-effector kinematic parameters

Robot Tool Centre Point (TCP) calibration is actually identification of its kinematic parameters and it presents an ordinary task which needs to be carried out to enable robot to perform any reasonable action with its end-effector, i.e., tool. Although repeatability of modern robotic manipulators is satisfying, they still need an information about the TCP in relation to their coordinate system. Besides manual method, robots can be calibrated in automated ways. The following discussion about identification of TCP parameters is mostly based on [68] [69] and the insights from designing the calibration unit [70].

One of the automated solutions is offered by ABB in form of BullsEye system [71]. Calibration unit resembles an arc with light beam generator and detector on its opposing ends. During calibration process, robot moves the tool inside the calibration unit intersecting the light

beam from different directions. Using this principle, 3-dimension TCP calibration can be performed.

Further improvement in performance, but using same principle was done by LEONI. Their advintec TCP-3D/5D unit is capable of up to 5-dimension TCP calibration. It uses two mutually perpendicular infrared beams, enabling it to increase number of dimensions it can calibrate, or decreasing the time and number of steps needed to perform calibration. Nevertheless, it still requires several steps to perform all the measurements, which increases calibration time.

One computer vision solution is presented in [72]. It uses an USB camera connected to a computer in order to acquire needed images. From those images, algorithms can perform a 6-dimension TCP calibration. To increase accuracy of this method, an iterative procedure was proposed by the author. While this approach offers a simple and affordable camera-based solution, it does not eliminate the effect of external influences, like lighting conditions or dust. Additionally, the algorithm looks for distinct points on the object, and can be confused when working with rotation symmetric tools.

The objective of this section is to describe an innovative camera-based calibration solution. Small area footprint and robust design of the calibration unit, explained in the following sections, enable easy installation in industrial environment and prove that it is practically orientated.

The main idea behind this calibration approach is to simultaneously acquire images of the tool from two orthogonal planes and use them to determine tool's orientation and TCP position. Robot needs to introduce the tool into the image acquisition area of the calibration unit, after which it will receive feedback in form of angular offsets in two planes from the reference tool orientation. After correcting angular offsets, the robot will receive information about TCP position offsets in *x*, *y* and *z* axis from the reference point in image acquisition area,.

Experimental validation section shows main observations based on initial test results. Data shows successful angle and position measurements and prove that the calibration principle can perform well with selected optics and hardware. However, it is noted that the best performance is achieved when the robot tool is positioned closer to the centre of the image, where camera distortion is least prominent.

It is concluded that presented principle offers a simple and reliable solution for up to 5-dimension TCP calibration. It also provides great flexibility and significantly reduces calibration and overall cycle time. This enables more frequent recalibration checks, which leads to higher quality of product. It is also mentioned that the design might have potential application in Factories of the Future. Conclusion also offers directions for further development and improvements.

4.3.1. Calibration Unit Design

In order to make the calibration unit suitable for industrial use, all the necessary components are packed into a single unit. At the top of the unit an image acquisition area is located, in a form of a box with open top from where the robot can introduce the tool. Two neighbouring vertical sides of the box are used as background panels and remaining two sides are transparent to enable image acquisition, as shown on Figure 34.

Background panels are illuminated with LED strips which can be triggered by camera. On the transparent sides of the acquisition area, two mirrors, placed at 45 degrees were introduced into system to reflect the image vertically downwards. As shown on Figure 35, cameras are positioned pointing upwards towards mirrors, close to the bottom end of corresponding vertical sides of calibration unit. By choosing this layout, small area footprint of the calibration unit was kept, while increasing the distance between cameras and the observed object, thus decreasing image distortion imposed by camera's lens. Mirrors are placed in sealed chambers to prevent contamination by dust or humidity. The camera used in the prototype is Basler acA2040-90uc USB 3.0 camera with the CMOSIS CMV4000 CMOS sensor. It is capable of delivering 90 frames per second at 4 MP resolution.

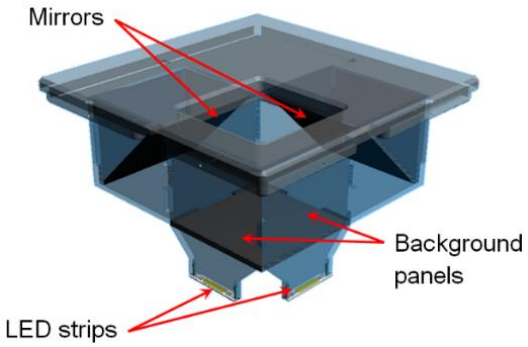

Figure 34. Image acquisition area with mirrors, background panels and LED strips. [68]

Power supply, fan-less PC, cables, and connectors are all located inside the calibration unit casing. Cover can be removed from one corner of the casing to enable easy access to internal components. The information about calibration unit is given in Table 6.

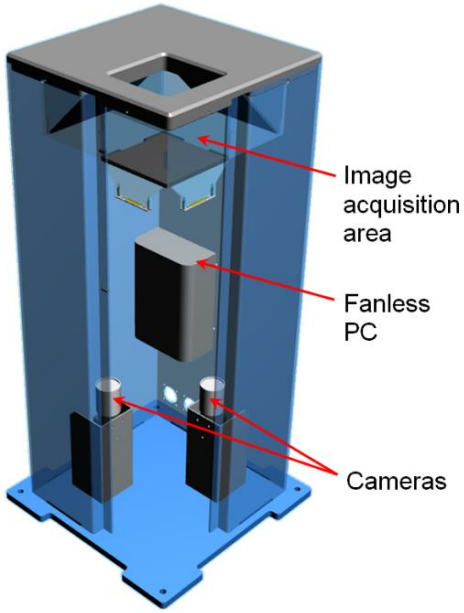

Figure 35. Overall layout of the calibration unit with access cover removed. [68]

Table 6 Calibration unit information. [68]

Camera type	Basler acA2040-90uc
Distance of camera from the acquisition area	353 mm
External dimension of the unit (l x w x h)	300 mm x 300 mm x 635 mm
Dimension of image acquisition area (l x w x h)	95 mm x 95 mm x 80 mm

4.3.2. Calibration Principle

Calibration of robot TCP means providing the robot with information about position and orientation of TCP in robot's coordinate system. Usually, it is required that provided information is relative to the robot's flange. Coordinates of flange are available at any moment from the robot's side, so what is really needed is only information about the position and orientation of the tool.

The main working principle for this calibration approach is based on acquiring images of the robot tool from two orthogonal planes. With properly illuminated background, a dark contour of the tool can be recognized by the pattern recognizing algorithm. From that point on, the calibration algorithm has two steps. First, the angle between the current longitudinal axis of the tool and previously referenced axis is measured on each image and forwarded to the control unit. If the current angle measurement is not satisfactory, robot can use provided information to make corrections in angle in order to match desired orientation. Second step is to measure the coordinates of the TCP on both image planes. Measured position in then compared with position of one point in space of image acquisition area, which represents desired position for the TCP. The offset between those two points is sent to the robot's control unit, in order to move the tool into desired position. Once the robot has guided the tool to match desired position and orientation, it remembers the position of its flange. Since coordinates of desired position in camera acquisition area are known, as well as coordinates of flange, parameters of the tool can easily be calculated and sent to the robot. The following text will provide the more detailed elaboration of the processes in described procedure.

The problem of identifying the tool and measuring the angle and position on the image is assigned to National Instruments' Image processing software, and it is not the subject of this discussion.

First issue needed to overcome is related with determining the angles of rotation. After the images have been acquired, and angles on both images have been measured, it is needed to determine the angles at which the tool must be rotated to reach desired orientation. Namely, out of two measured angles, only one can be considered to be actual angle, in which case the second angle is the consequence of perspective distortion. If the goal is to lead the robot tool into desired angular position using two sequential rotations in corresponding image planes, then it is of crucial importance to determine the exact value of both angles, excluding the perspective distortion.

Perspective distortion of one angle value is a result of the fact that one of the rotations is not in plane that is parallel to the corresponding image plane. Let us consider that the measured angular position of the tool is a result of two sequential rotations in mutually orthogonal planes.

As mentioned previously, planes of two images are also orthogonal. Suppose right-handed Cartesian coordinate system is used, with its origin in TCP and with *z* axis pointing vertically upwards. Further, let us consider that one image plane is parallel to *xz* plane, and the other is parallel to *yz* plane, as shown on Figure 36.

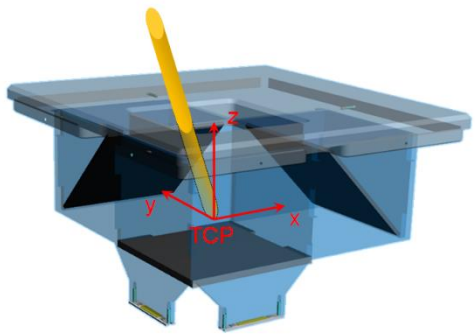

Figure 36. Image coordinate system within image acquisition area. [68]

Without losing generality, we can consider that the initial orientation of the tool is vertical, and that it has length *l*. Supposing that the first rotation is about axis parallel to *x* axis, the angle of rotation α is visible in *yz* plane, and the *y* axis projection of the object length, l'_y is equal to (19):

$$l'_y = l\,sin(\alpha) \tag{19}$$

In *xz* plane, the object would appear vertical, but shorter than before rotation. If the rotation angle about *x* axis is α, then the apparent height of the object in *xz* plane, l', is equal to apparent projection of the tool length to *z* axis l'_z (20),

$$l' = l'_z = l\,cos(\alpha) \tag{20}$$

Next, the tool rotates at an angle β about *y* axis, and this action affects views in both planes. In *xz* plane, the rotation angle appears to be β, and the length of the tool remains apparently unchanged. In *yz* plane, the length of the object appears to be shorter. However, the *y* axis projection is unaffected, as written in (21). The *z* axis projection of the length after second rotation is equal to (22).

$$l''_y = l'_y \tag{21}$$

$$l''_z = l'_z\,cos(\beta) = l\,cos(\alpha)\,cos(\beta) \tag{22}$$

It is now clear that the apparent angle in *yz* plane is also affected by the second rotation, it is now bigger than α, and it is equal to (23):

$$\alpha' = arctan\left(\frac{l''_y}{l''_z}\right) = arctan\left(\frac{sin(\alpha)}{cos(\alpha)\,cos(\beta)}\right). \tag{23}$$

Angles that would be seen by the pattern recognition algorithm are α' in *yz* plane and β in *xz* plane. The angle of second rotation remained unaffected by the perspective distortion. In an analogous way, it can be proved that if the rotations happened in different order, the angle α would stay unaffected, and the angle β' would be (24):

$$\beta' = arctan\left(\frac{sin(\beta)}{cos(\alpha)\,cos(\beta)}\right). \tag{24}$$

Having shown that the angles seen on image are not the real angles of rotation, but result of perspective distortion, it is needed to find a way to calculate real angles. In reality, the algorithm would not know which rotation happened first. Therefore, for every pair (α_{meas},β_{meas}) of measured angles in *xz* and *yz* plane respectively, two pairs of real angles exist, (α_{meas},β) and (α,β_{meas}).

According to (23), if the first rotation happened about *x* axis, the real angle α is not visible, but it can be calculated using (25).

$$\alpha = arctan(tan(\alpha')\,cos(\beta)) \tag{25}$$

In similar way, formula for calculating angle β can be derived from (24).

Having determined the real angle of the rotation, it is possible to position the tool vertically. Also, it is clear that by rotating the tool for the opposite angles, it can be brought to the initial position from vertical position. The situation where desired tool orientation is not vertical can be solved in different ways. One of the trivial solutions is to first measure the angles in desired position and calculate real angles of rotation. In that way, it is known how to move the tool from desired position to the vertical position, and vice versa. Then, for any other measured position, tool can be first brought to vertical position, and from that position rotated for the angles that bring it to the desired position.

The process described above represents the first step of calibration procedure. Having orientated the object correctly, it is important to measure the translation offset of the TCP from the desired position of TCP. This operation is first done by measuring and remembering the desired TCP coordinates on both images, and then comparing it to the TCP position of the tool in every calibration check. If the measured TCP position does not match the desired position, robot can correct it using information provided by the calibration unit. Then, the robot receives TCP parameters calculated by the calibration unit based on flange coordinates when TCP is in desired point, and previously known position of that desired point.

Theoretically, the calibration procedure can be performed without the movement which compensates the position. The solution would be to use offset information and flange coordinates in order to calculate what would be the flange coordinates when TCP is in desired position. However, this action is not recommended because the accuracy of offset measurement cannot be guaranteed in every point of image acquisition space due to image distortion.

Because of camera distortion, formula given with (25) is applicable in situations where cameras with telecentric lenses are used. However, the size of telecentric lenses that can cover the needed field of view and their price are far too big to be practically useable in industrial applications. For example, the length of telecentric optics which have field of vision of 62 mm in diameter is more than 540 mm in length, which combined with other components would give total height of the calibration unit of more than one meter.

When using regular lenses, effects of camera distortion become less prominent as the distance from the observed object and camera increases, and theoretically decrease to zero as the distance increases to infinity. In order to increase the distance of the camera from the tool, cameras are positioned in the way explained previously within calibration unit design section. The described solution is a result of the trade-off between calibration unit dimensions and desirable distance.

4.3.3. Image acquisition results and discussion

Testing consisted of two phases. In first phase, image acquisition and pattern recognition algorithm were tested separately. The calibration unit was tested to check if the images are acquired with good quality and reliability, Special attention was dedicated to external influences analysis, which proved to be not significant.

Pattern recognition algorithm and (25) were first tested with images generated using a 3D modelling and rendering software. The advantage of these images is that the virtual tool can be rotated at any known angle, so the (25) can be tested to full extent. These tests also showed that the pattern recognition works accurately, and reliably. Angles obtained from the algorithm were measured with accuracy of under 0.1 degrees. Calculations of distorted angle matched those of the real rotation with similar accuracy.

Second phase was dedicated to testing the whole system, with real test objects and real environmental influences. Based on initial tests, it was concluded that the distortion effect is not significantly prominent and that (25) can be used with good results.

Pattern recognition algorithm proved itself to be very reliable, with high repeatability, and good precision. For testing and development purposes, several functionalities were introduced into program. One of them checks if the algorithm has measured the angle correctly. It does so by rotating the image for the angle determined by the algorithm, but in the opposite direction. In that way, the operator can visually check if some notable mistake happened. The angle measuring algorithm can be run on the rotated image to check if the angle is equal to zero. This functionality is shown on Figure 37 and Figure 38. Image acquired from one camera with test object is shown on Figure 37. The angle measured by recognition algorithm was -19.1º from vertical plane. Section of the same image, rotated for measured angle, is shown on Figure 38. Shown pictures and tests demonstrate that the pattern recognition algorithm has high precision in determining angular orientation.

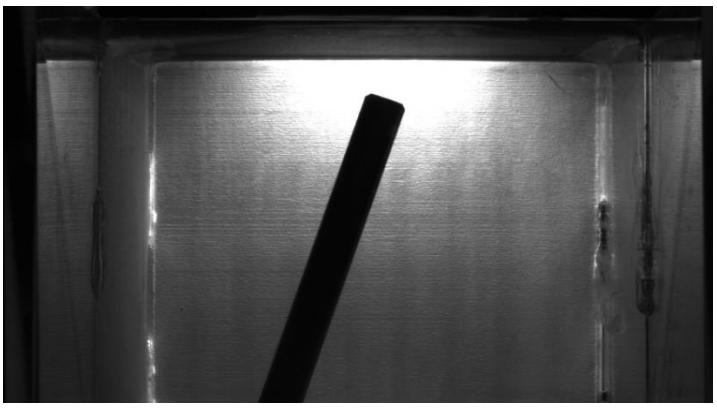

Figure 37. Image obtained by camera in image acquisition area. [68]

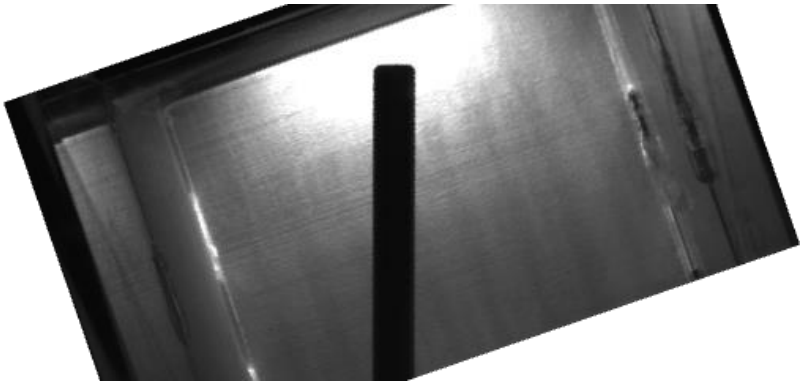

Figure 38. Image from Fig. 4, rotated for 19.1º. [68]

Some of the introduced functionalities are shown on Figure 39. This image demonstrates three tests. First one, shown on upper right part of the image shows the TCP, as recognized by the algorithm. On the image, the TCP is represented by the centre of the red circle. Second test is performed on lower left part of the Figure 39. Area of the image painted in red shows which part of the image is recognized as part of the tool. The third check is visible on both upper and lower part of the Figure 39 in form of a straight blue line. The line goes through the centre of the tool, as determined by the algorithm.

These tests were performed on a number of test objects. Some of test objects were designed especially for testing purpose, such as one shown on Figure 37 and Figure 38, while some of them are every-day objects, such as screw shown on Figure 39. All objects were held in place by a specially designed plate with drilled holes through which the test objects can be inserted into image acquisition area. The plate can be rotated to check if the algorithm measures angles symmetrically, and do both cameras see the same angle. For example, the angle measured on Figure 37 was 70.9º from the horizontal plane, while the angle on Figure 39 was 109.1º, which means that both of them are 19.1º in absolute value from the vertical plane. This demonstrates that the algorithm does not favour different areas of the image, and that the camera is positioned correctly, i.e., the camera does not introduce any angular error. Tests for camera in different plane were done with similar conclusion.

One of the testing phases also included testing of TCP identification in situations where it is not clearly visible. It can happen that, due to unfavourable orientation of the tool, the real TCP is not visible for the algorithm, as it is shown on Figure 40 on example A. Therefore, it must be estimated. At this moment, such functionality of pattern recognition algorithm is not developed. However, the longitudinal axis of the object can be recognized, and therefore tool can be oriented in a favourable way, in which the TCP can be properly identified.

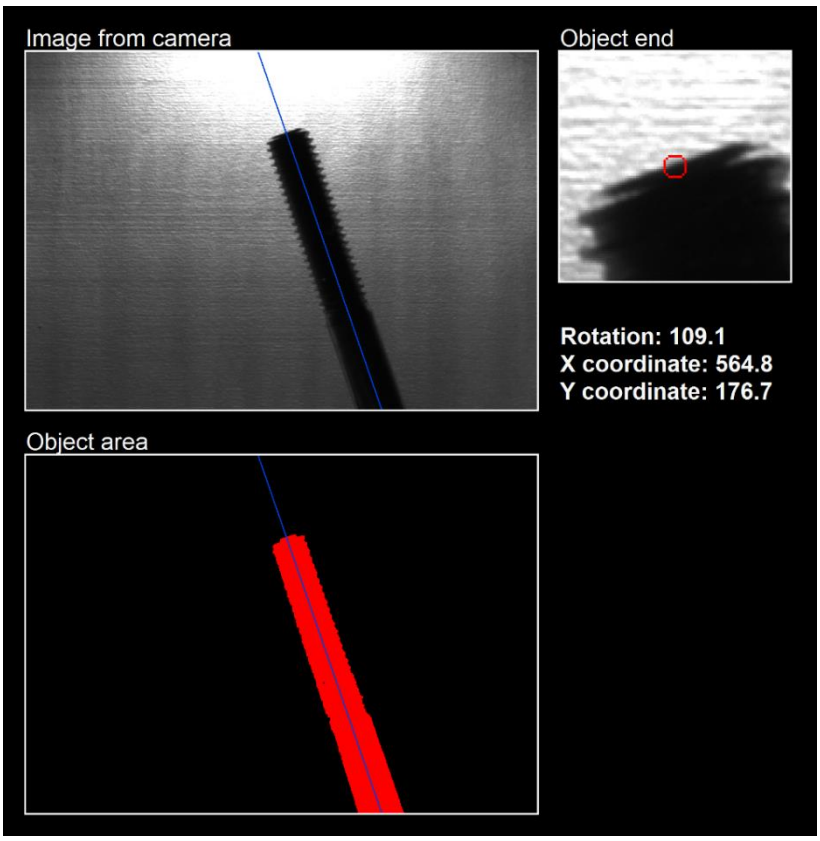

Figure 39. Images used for algorithm verification. [68]

During testing, it was noticed that the distortion is less prominent in the areas closer to the centre of the image. Also, it was noted that the angular measuring precision is greater if the tool is closer to vertical position. Therefore, in order to have optimal conditions regarding accuracy, it is recommended to set desired orientation to vertical, and desired TCP position in the centre of images, as it is shown on Figure 40 on example B.

Pattern recognition was tested for robustness by deliberately damaging the shape of the images. This checks how the algorithm would behave in situations where tool is deformed, or the lighting conditions on image are not optimal. One other thing that was recognized as a potential source for errors were scratches, dirt specs, and other types of visual contaminations on transparent sides, or on background panels in the image acquisitions box. All listed disturbances were not able to introduce any significant error, thanks to the fact that the pattern recognition algorithm observes whole object, rather than only one section of it. This presents an important advantage compared to the methods where moving tool intersects light beams.

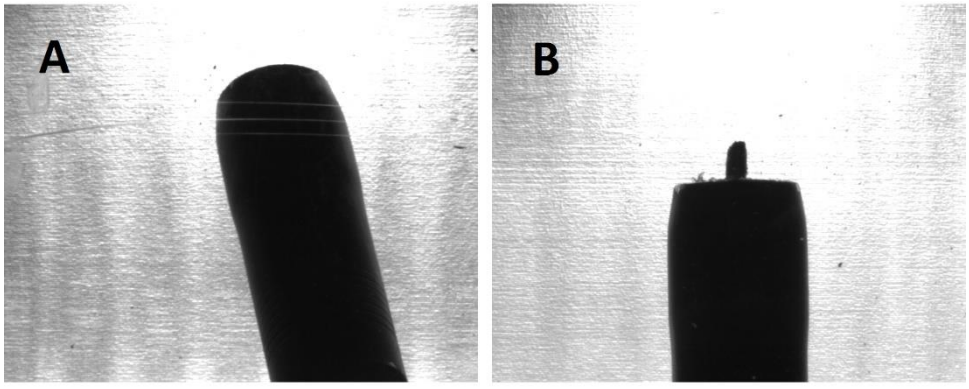

Figure 40. Test object: A - unfavourable orientation; B - favourable orientation. [68]

4.3.4. Implementation

Local Coordinate System

In order to guide the tool according to the information obtained from analysed images, it is necessary to establish a relation between the coordinate system defined by cameras, and the coordinate system of the robot's base. In essence, a transformation matrix between the two coordinate systems must be obtained using only the calibration unit and any tool that can fit its TCP within the image acquisition area.

The cameras' coordinate frame consists of the horizontal axes of two orthogonal images which form its *x* and *y* axes, while the vertical axis is represented by the *z* axis. If the coordinates of a point in the reference frame of robot's base are $C_b=[x_b\ y_b\ z_b]^T$, and coordinates of the same point in reference frame of cameras $C_c=[x_c\ y_c\ z_c]^T$, the relation (26) connects them.

$$C_c = T_{cb}(C_b - V_{cb}), \tag{26}$$

where T_{cb} is a matrix that performs orientation alignment of axes, and V_{cb} is a column-vector that represents the translational offset between origins of the two frames.

When the origins of two reference frames are matched, by analysing the transformation matrix, it is possible to identify that columns in matrix T_{cb} represent projections of respective unit vectors in robot's the base frame onto axes in the cameras' frame, as shown in (27). First column represents projections of $[1\quad 0\quad 0]_b^T$ onto *x*, *y* and *z* axis of cameras' frame, second column of $[0\quad 1\quad 0]_b^T$, and third of $[0\quad 0\quad 1]_b^T$, where subscript *b* denotes that these are coordinates of unit vectors in the reference frame of robot's base.

$$T_{cb} = [[1\quad 0\quad 0]_b^T proj.\quad [0\quad 1\quad 0]_b^T proj.\quad [0\quad 0\quad 1]_b^T proj.] \tag{27}$$

With this perspective in mind, it is possible to perform unit movements of the robot along each axis of its coordinate system and observe their projections in coordinate system of cameras. The idea is to assume that the first point where the robot's TCP is located in the cameras' frame represents the origin of coordinate system that is aligned with cameras. Then, upon performing unit movement of the TCP along the *x* axis of the robot's frame, it is possible to measure the offset between the new position and the position of the origin in each axis. The offsets represent projections of the first unit vector onto the axes of coordinate systems of cameras and give the first column of the transformation matrix. A similar procedure is applied for unit movement along *y* axis and second column, and along *z* axis for third column of the transformation matrix T_{cb}.

From the implementation point of view, it is important to determine which distance should be considered as a unit movement. Although most logical, movement of 1 millimetre is not big enough because of limited resolution of cameras, and accuracy issues of the robot. The solution is to perform longer movements, and then normalize the obtained results.

Using the acquired matrix T_{cb}, it is possible to determine the Euler angles of transformation between two coordinate systems, needed by the robot's controller to define a new coordinate system. For this purpose, it is necessary to know the order of elemental rotations, and in this case, the X-Y-Z order was used. Therefore, matrix T_{cb} is equal to matrix *T* containing Euler angles R_x, R_y and R_z and given by (28)-(32).

$$T = \begin{bmatrix} cos(R_y)\,cos(R_z) & cos(R_y)\,sin(R_z) & -sin(R_y) \\ T_{2,1} & T_{2,2} & cos(R_y)\,sin(R_x) \\ T_{3,1} & T_{3,2} & cos(R_x)\,cos(R_y) \end{bmatrix} \tag{28}$$

$$T_{2,1} = cos(R_z)\,sin(R_x)\,sin(R_y) - cos(R_x)\,sin(R_z) \quad (29)$$

$$T_{2,2} = cos(R_x)\,cos(R_z) + sin(R_x)\,sin(R_y)\,sin(R_z) \quad (30)$$

$$T_{3,1} = sin(R_x)\,sin(R_z) + cos(R_x)\,cos(R_z)\,sin(R_y) \quad (31)$$

$$T_{3,2} = cos(R_x)\,sin(R_y)\,sin(R_z) - cos(R_z)\,sin(R_x) \quad (32)$$

The Euler angles can be calculated using (33)-(35) obtained from (28):

$$R_x = arctan(T_{2,3}/T_{3,3}) \quad (33)$$

$$R_y = arcsin(-T_{1,3}) \quad (34)$$

$$R_z = arctan(T_{1,2}/T_{1,1}). \quad (35)$$

The transformation matrix T_{cb} and angles obtained can be used to transform coordinates from the robot's frame to the cameras' frame. Since the transformation matrix T_{cb} is real, symmetric, and orthogonal, the inverse matrix is equal to transposed matrix. Therefore, in order to transform from cameras' to robot's base reference frame, expression (36) can be used:

$$C_b = {T_{cb}}^{-1} C_c + V_{cb} = {T_{cb}}^{T} C_c + V_{cb} \quad (36)$$

The described procedure offers the theoretical background for acquiring matrix T_{cb}, therefore performing alignment of coordinate systems of the camera and robot's base. However, it is not described how to obtain the offset vector V_{cb}.

Due to the fact that the tool used to perform alignment is unknown, the coordinates of the TCP seen on images are also unknown, and the whole procedure is performed using coordinates of the robot's flange. While the axes of the coordinate system defined in this way are parallel to the respective axes of cameras' reference frame, the offset introduced by the tool between them is not compensated. The next section will provide the means for calculation of the complete offset V_{cb}.

The procedure was designed such as to enable alignment using any tool that can fit its tip into the image acquisition area, making it easy to implement and adapt without any changes to the system. The procedure needs to be executed only once, when the relative position of the robot and calibration unit are determined and fixed. All the movements and calculations are performed automatically, so the operator only needs to guide the tool into the image acquisition area, and run the procedure.

Initial Calibration of the TCP

Once the alignment of coordinate systems is performed, the TCP can be moved along the axes of the cameras frame at any desirable distance in any direction. However, at this point, the origin of the newly defined reference frame (and therefore coordinates of points within image acquisition area) is unknown in the robot's base reference frame. This section will describe the TCP calibration procedure which does not require any prior information about the tool, and which will later be used to acquire coordinates of points inside the camera's reference frame, and the vector V_{cb}.

In order to define a tool, it is usually required to define a 6-dimension vector that contains positions and orientations of the TCP relative to the coordinate system of the robot's flange. One solution of determining tool parameters without any previous information is to guide the TCP in the same point several times, from different directions. In theory, it is enough to approach the same point from only two directions, but they need to be different at least in two angles. In that way, it is possible to find position coordinates of the TCP relative to the flange. Position and orientation coordinates of the flange in both positions are known, since they can be obtained from the robot's

controller, and the point TCP_b, where the tool's tip is guided, is the same in both cases. By assuming that the tool is rigidly fixed to the flange, it is possible to write (37):

$$T_1^{-1}TCP_f + V_1 = T_2^{-1}TCP_f + V_2 = TCP_b \tag{37}$$

Subscripts *f* and *b* of TCP signify that these are TCP coordinates in the reference frames of the robot's flange and base, respectively; T_1 and T_2 are transformation matrices between the coordinate system of robot's base and flange in the first and second position, and V_1 and V_2 are offsets between the reference frames of the robot's base and flange in these two positions. T_1, T_2, V_1, V_2 are all known parameters provided by the robot's controller in form of 6-dimension vector of the flange's positions and orientations. *T* matrices are acquired by substituting respective R_x, R_y and R_z angles of the flange orientation into (28), and *V* matrices consist of flange offsets from the origin of the robot's base coordinate system, $V=[x_{offset}\ y_{offset}\ z_{offset}]^T$.

An analytical solution to the problem of finding the TCP position can be obtained using (13). However, in practice this solution requires very high precision, and it is highly unlikely that it would work mainly because of two reasons. The first reason originates from inaccuracies of the robot's structural parameters which lead to direct kinematics inaccuracy [73], [74] which means that position and orientation information obtained from the robot is usually not accurate enough. The second reason is that it is very difficult to guide the TCP in exactly the same point due of finite resolution of measuring and guiding devices. These two issues, although irrelevant in most practical applications, can lead to a situation where two directions do not intersect, and therefore the solution does not exist. In reality, the solution would be to take the points where direction axes are closest to each other.

A suboptimal solution can be implemented using a numerical solving algorithm. Although in theory it could work with only two points, it is advised to use at least four points for improved accuracy, (Figure 41 a). The solution is a point TCP_f from the search region R_f in the flange's coordinate system, for which the sum (38) is the smallest.

$$\sum\left(T_iTCP_f - T_jTCP_f\right)^2, i = \overline{1,n} - 1, j = \overline{i,n}, TCP_f \in R_f \tag{38}$$

It is evident that the search region R_f (Figure 41 b) must be big enough to enable wide variety of tools, but the algorithm must also provide sufficient accuracy. Time and computational resources needed to meet these two conditions at the same time are very high. The solution is in an iterative algorithm, which uses relatively low search resolution in the first iterations, when the search region is big, and gradually increases the resolution while narrowing the search region as iterations progress.

The process of guiding the robot into the same point from several directions is done by the vision system, being enabled by previously performed axes alignment. Once at least four flange coordinates are obtained, the calibration unit performs the calculations using the previously described principles and defines the TCP position.

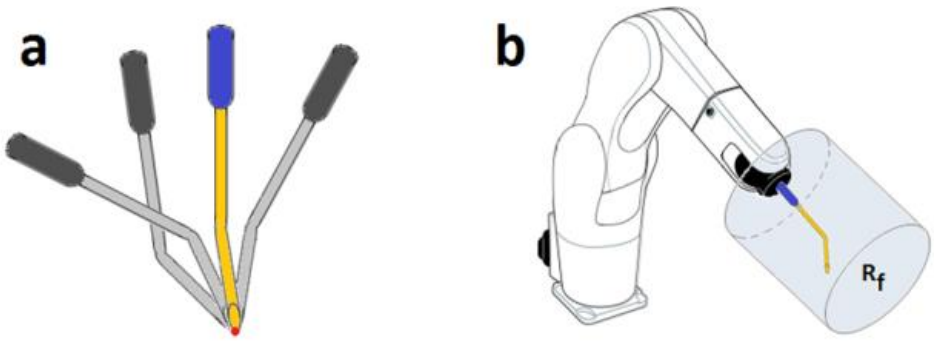

Figure 41. Example of positions for TCP calibration (a) and search region R_f (b) [68]

Using the tool defined with this procedure, the coordinates of any desired point in image acquisition area can simply be requested from the robot and used as vector V_{cb}.

From the perspective of obtaining DH parameters of the tool, i.e., the robot's end-effector, at this point it is possible to calculate them using previously described procedure for the identification of DH parameters of the robot itself. Namely, since position of the TCP is known in the coordinate system of camera as well as the orientation of the tool's longitudinal axis, these values can be transformed into coordinate frames of the robot's base, or flange. The longitudinal axis of the end-effector is conventionally regarded as its *z* axis, and the known TCP position completes the pair of parameters needed to define the joint vector, or joint rotation vector, as regarded in the section 4.2. From that point on, the procedures described in section 4.2 can successfully be used to obtain DH parameters of the end-effector and include them in the overall DH model.

Fast Calibration Procedure

This procedure practically implements the calibration principle and algorithm explained in this section and in [69], since all its preconditions are fulfilled. Tool parameters are typically coordinates of the TCP in the robot's flange coordinate frame. If the TCP is guided into a point in space whose coordinates are known the robot's base frame, and the position and orientation of the robot's flange are also known in the same coordinate system, the tool parameter calculation is a matter of transformation of coordinate systems (39):

$$TCP_f = T(TCP_b - V), \tag{39}$$

where TCP_b represents the position of the TCP in the robot's base frame, V stands for position coordinates of the robot's flange in its base frame, and TCP_f are coordinates of the TCP in the coordinate system of robot's flange.

Evidently, this procedure requires far less time and computational resources than the previous one. It is used in regular operation, and it performs two main tasks. The first task is to check whether the position and orientation of the tool are within tolerances relative to the reference position when the flange is in its reference position. If the result is true, the robot will continue its normal routine. The second task is to orientate and guide the tool to the reference position and orientation if it is not there and recalibrate it using the new position of the flange, after which the robot can again proceed with normal operation. The design of this technique additionally simplifies the implementation since the same routine verifies and corrects the tool parameters. As a result, the TCP is always calibrated and the robot can resume with its regular operation after running this procedure.

4.3.5. Algorithm Observations and Discussion

The evaluation of the calibration principle was performed and presented earlier, and it is not in focus of this section. The recommendations to set the reference position of the TCP close to centre of the image acquisition area and the reference orientation close to the vertical axis in both images, should be followed to reduce distortion and achieve optimal performance.

The procedure for axis alignment, as well as other procedures, was tested using a DENSO VS-087 6-axis industrial robot and information obtained from the camera system. After the alignment was performed, the movement was attempted along axes of the camera system. Although the axes were aligned with adequate precision, the results of the test also demonstrated that the movement on longer distances resulted in offsets of less than 2% of attempted distance in axes that should have remained unchanged. A greater positioning accuracy can be achieved by introducing positioning check. If the position recorded after the movement is not within set tolerances, additional correction can be performed. It is reasonable to set positioning tolerance to be at least as big as the robot's declared repeatability. The robot used for testing has very good repeatability of 0.02 mm. Considering the maximum percentage of error, and the size of the image acquisition area,

in order to guide the TCP within tolerances, no more than two movements must be performed. The main reason for this positioning imperfection lays in the fact that the image is not orthographic, but slightly distorted because of the finite distance between the camera and the observed object. However, this effect is only noticeable in areas that are close to the borders of the image acquisition area and does not affect the robot's guidance in regular operation. The procedure was repeated for various relative orientations between the calibration unit and the robot, and conclusions were consistent.

The procedure for initial calibration of the TCP was tested in order to verify its accuracy. The first step was to compare its performance with the performance of built-in TCP calibration functionality of the Denso robot. The functionality uses a similar principle, and requires identical type inputs, being therefore suitable to compare them. When identical coordinates were used for both algorithms, the difference in obtained tool parameters was less than 0.1 mm. The second testing phase was to identify potential sources of inaccuracy. The procedure was performed for various angles, and it was concluded that calibration results were more accurate when angles of rotation were bigger.

However, it is not recommended to perform big angle rotations because of high risk of collision with the calibration unit. The reason is that at the start of the procedure, the robot has no information about the tool, and therefore rotations have to be made around the centre of the robot's flange. In that way, if the tool is long, even small angle rotation would result in big offsets of the tip of the tool. The solution is to perform two sequential calibrations. The first calibration uses small rotation angles, and therefore does not result in big movement of the tool's tip, but results in lower accuracy. However, the accuracy is good enough to provide initial information about the tool parameters. It is possible and recommended to repeat the procedure with bigger angles, but this time around previously provided TCP information. The result of the second iteration is calibration with higher accuracy without the risk of collision with the calibration unit. Testing was performed with various shapes and sizes of tools, which proved that the initial calibration of the tool can be successfully performed in two steps with different tools.

The procedure for fast calibration uses the result of axes alignment and initial calibration to check and correct the tool parameters. In this procedure, a position tolerance is set to consider small deviations of TCP's position caused by minor tool deformations or robot's repeatability issues. The procedure was tested with different levels of distorted tools, ranging from untampered tool to realistic impact scenario where both position and posture of the tool have been changed. In some tests, changes were made to the extent where such deviations would indicate serious tool damage.

However, in all cases the procedure was performed according to expectations and managed to recalibrate all the necessary parameters. In order to ensure correct positioning in all circumstances, a check was also introduced to verify whether the position and orientation were properly corrected. If they were not, an additional correction iteration is going to be performed.

The intention of this section was to present theoretical and practical concepts and procedures implemented during the development of a solution for robot TCP calibration. The idea to use images from orthogonal planes to acquire TCP information, conceived in earlier phases of research, served as a basis for development of calibration unit. The presented algorithms and routines serve as crucially important links between theoretical concepts and practical realization ready for implementation in industry. The procedure for axes alignment uses a simple and reliable algorithm to enable the calibration unit to guide the robot's tool. The routine used for the initial calibration of the tool additionally serves as a mean to determine coordinates of the points within the image acquisition area and therefore enables a faster algorithm to be used in subsequent tool identifications or TCP checks in regular operations.

The fast calibration procedure is the only one used in regular operation, and it can perform checks of proper calibration, or recalibrate the tool if its parameters have changed. The important benefit of this procedure it that its outcome is always a calibrated tool, which means that the robot can continue with regular operation without human-assisted recovery. This is especially important

in environments where tools are frequently switched using an automatic changing station in order to perform different operations.

From the beginning, the system was practically oriented, and therefore its construction and all the algorithms were designed to be accurate, reliable, and robust. All procedures are independent from other devices, and none of them requires any modifications of the system, or some specially developed tool, which makes them easy to implement. Human intervention is only needed to guide the new tool into the unit and run the procedure during the installation phase, or when introducing an entirely new tool. This means that the system is completely automated, and therefore it can be integrated in facilities that rarely require human presence. The short time needed for calibration checks means they can be performed more frequently, which ensures higher quality of production. From the versatility point of view, one calibration unit can be used by multiple robots, if it is in their working range, and it is universal to all robot brands.

4.4.Kinematically Augmented mDTW

The first step in the inclusion of kinematics-related parameters into modified Dynamic Time Warping (mDTW) is a determination of which parameters would be most effective in reflecting the effects of the changes in spatial relation of robot joints, which affect the projection of external forces. The projection of external forces originating from the contact task onto axes of interest can be determined reliably using a kinematic model of the robot. For this reason, it is necessary to identify the kinematic parameters of the robot itself as well of its tool, i.e., end-effector, as presented in sections 4.2 and 4.3. Denavit–Hartenberg (DH) notation is one method that is commonly used for this purpose and, as such, is considered for the representation of the kinematic parameters used in the algorithm. Still, the idea is also valid for any other notation. The DH model is represented in the form of homogenous transformation matrices H.

Once the DH parameters of the entire robot and end-effector have been obtained, it is possible to use them to determine projections of the contact force profile onto the individual joint of the robot. To that end, it is first necessary to determine the relationship between the joint of interest and the end effector frame. This is achieved by multiplication of matrices H_{i-1}^{i} from the observed joint o to the end effector indexed with e (40) and (41):

$$H_{i-1}^{i} = \begin{bmatrix} \cos(\Theta_i) & -\sin(\Theta_i)\cos(\alpha_i) & \sin(\Theta_i)\sin(\alpha_i) & a_i\cos(\Theta_i) \\ \sin(\Theta_i) & \cos(\Theta_i)\cos(\alpha_i) & -\cos(\Theta_i)\sin(\alpha_i) & a_i\sin(\Theta_i) \\ 0 & \sin(\alpha_i) & \cos(\alpha_i) & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{40}$$

$$H_o^e = \prod_{i=o}^{e} H_i. \tag{41}$$

Based on the calculated matrix H_o^e, it is possible to determine the distance between joint of interest and the Tool Centre Point (TCP) in coordinate frame of the observed joint of interest o (42):

$$l_o^e = (x_{oe}, y_{oe}, z_{oe}) = (H_o^e(1,4), H_o^e(2,4), H_o^e(3,4)) \tag{42}$$

The calculated distances in the relevant joint coordinate frame effectively represent the lever lengths l_o^e that may affect the torque in that joint. Since for single-degree-of-freedom (DoF) revolute joints the only axis affected by torques is the *z* axis, the actual lever components that may affect the torque are those perpendicular to the *z* axis. This means that only x and y components of the lever l_o^e are relevant for further calculations, and therefore, they are the only ones depicted in Figure 42 in blue.

If the normalized components of contact force are known in the robot base coordinate frame, F_b, they first need to be transformed into the coordinate frame of the tool (43)–(45):

$$H_b^e = \prod_{i=1}^{e} H_{i-1}^i \tag{43}$$

$$T_b^e = \begin{bmatrix} H_b^e(1,1) & H_b^e(1,2) & H_b^e(1,3) \\ H_b^e(2,1) & H_b^e(2,2) & H_b^e(2,3) \\ H_b^e(3,1) & H_b^e(3,2) & H_b^e(3,3) \end{bmatrix} \tag{44}$$

$$F_e = F_b \cdot T_b^e, \tag{45}$$

where H_b^e is the homogenous transformation matrix from the coordinate frame of the robot base into the coordinate frame of the tool, and F_e represents components of the force in the end-effector coordinate frame, as shown in the upper right part of Figure 42 in green colour.

If the normalized components of the contact force F_e are known in the coordinate frame of the tool, steps (43)–(45) can be skipped. Once the relevant kinematic chain has been calculated, it is then possible to determine its components in the coordinate frame of the joint of interest (46) and (47).

$$T_o^e = \begin{bmatrix} H_o^e(1,1) & H_o^e(1,2) & H_o^e(1,3) \\ H_o^e(2,1) & H_o^e(2,2) & H_o^e(2,3) \\ H_o^e(3,1) & H_o^e(3,2) & H_o^e(3,3) \end{bmatrix} \tag{46}$$

$$F_o = F_e \cdot T_o^{e\prime} \tag{47}$$

The cross product of relevant lever length components and components of the contact force in the tool coordinate frame yields components of torques in the tool coordinate frame, while the absolute value of the *z* component of the cross product, shown in purple in Figure 42, is equal to the equivalent lever length for the joint of interest (48).

$$l_{eqiv} = |(l_o^e \text{ x } F_o)\ [0\ 0\ 1]| \tag{48}$$

Calculation of the lever lengths for each sample contained in the measurement vector and reference signal enables morphing both the reference value and the measurements to enable better comparison. When values of the measured samples, i.e., deviations from the nominal values, and corresponding lever lengths are observed, it is evident that there is a significant correlation. Depending on the dynamics of the measurement signal caused by the robot's movement on the observed sequence, type of task, and observed joint, this correlation coefficient $\rho \in [-1,1]$ ranges from 0.64 to 0.95, while a simple smoothing filter raises the lower limit to 0.84, confirming the intuitive assumption.

At this point, it is important to make a physical interpretation of the connection between the calculated equivalent lever lengths and the deviations in the measurement vector. Deviations within the measurement vector correspond to torques produced by the unmodelled external contact force acting on the equivalent lever length. The intensity of force components does not have to be known explicitly, since the effect of components on an individual joint axis is implicitly reflected through intensities of deviations.

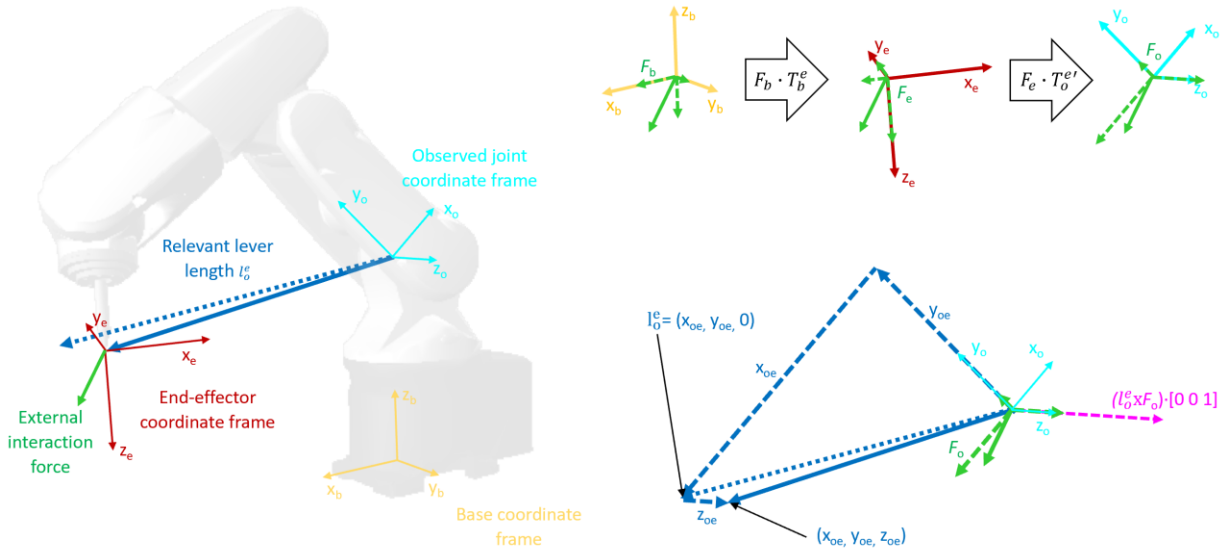

Figure 42. Coordinate frames and transformations [59]. The left section of the figure illustrates the base and tool coordinate frame as well as the coordinate frame of the observed joint together with levers and direction of the external interaction force. The upper right section illustrates the steps of transforming the external force into the coordinate frame of the observed joint. The lower right section of the figure shows the distance between the origins of coordinate frames of the observed joint and the end effector projected onto the x-y plane to determine the relevant lever length. This section of the figure also shows the relevant component of the torque, whose absolute value equates to the equivalent lever length.

From the implementation perspective, to enable calculation of the lever lengths, the measurement signal received from the robot must contain measurements of joint angles, in addition to the measurements of joint currents or estimated torques, which are used to indicate the presence of contact. For each joint, the measurement vector $M_v(i)$ contains two pieces of information, joint current or estimated torque $a(i)$ and rotation angle from the encoder $\Theta_v(i)$ (49). Similarly, measurements based on which the reference sequence $M_r(j)$ was stored contain components of joint current or estimated torque $b(j)$ and rotation angle $\Theta_r(j)$ (50).

$$M_v(i) = \big(a(i), \Theta_v(i)\big), 1 < i < m \tag{49}$$

$$M_r(j) = \big(b(j), \Theta_r(j)\big), 1 < j < n \tag{50}$$

Inclusion of lever lengths into the mDTW is done with the substitution of Rules (8)–(11) with Rules (51)–(54).

$$d(1,j) = \left|\frac{a(1)}{l_a(1)} - \frac{b(j)}{l_b(1)}\right| + \min\left(\left|\frac{a(2)}{l_a(2)} - \frac{b(j+1)}{l_b(j+1)}\right|, \left|\frac{a(2)}{l_a(2)} - \frac{b(j)}{l_b(j)}\right|, \left|\frac{a(1)}{l_a(1)} - \frac{b(j+1)}{l_b(j+11)}\right|\right), i = 1, j \neq n \tag{51}$$

$$d(1,n) = \left|\frac{a(1)}{l_a(1)} - \frac{b(n)}{l_b(n)}\right| + \left|\frac{a(2)}{l_a(2)} - \frac{b(n)}{l_b(n)}\right|, i = 1, j = n \tag{52}$$

$$d(i,1) = \left|\frac{a(i)}{l_a(i)} - \frac{b(1)}{l_b(1)}\right| + d(i-1,1), 1 < i \leq m, j = 1 \tag{53}$$

$$\mathrm{d(i,j)} = \left|\frac{\mathrm{a(i)}}{\mathrm{l_a(i)}} - \frac{\mathrm{b(j)}}{\mathrm{l_b(j)}}\right| + \min(\mathrm{d(i-1,j-1), d(i,j-1), d(i-1,j)}), 1 < \mathrm{i} \leq \mathrm{m}, 1 < \mathrm{j} \leq \mathrm{n} \tag{54}$$

Within these rules, l_a(i), $1 < i \leq m$ represents the equivalent lever length calculated for *i*-th sample of the measurement vector using $\Theta_v(i)$, whereas $l_b(j)$, $1 < j \leq n$ represents the equivalent lever length calculated for *j*-th element of the reference sequence using angle $\Theta_r(j)$, for the observed joint using (40)–(48). Effectively, while forming the matrix d, each of elements $a(i)$ and $b(j)$ are divided by their corresponding lever lengths calculated for the observed joint based on the measurements of the joint angles, whose values are stored as part of the measurement vector and reference sequence, respectively. This modification does not affect the causality since the manner in which the elements of matrix d are formed is unchanged. The search algorithm can still end at any element in the first row, although it may no longer be the same element as in the mDTW.

Rules (55) and (56) for searching the optimal path in matrix d are the same as Rules (12)–(13) of the mDTW:

$$min(d(i-1,j-1), d(i,j-1), d(i-1,j)), i \neq 1, j \neq 1 \tag{55}$$

$$d(i-1,j), i \neq 1, j = 1 \tag{56}$$

The search will start from the element in the last row that has the minimal sum, $min(d(m,j))$, $1 \leq j \leq n$, though again, it may not be the same element as with the mDTW. Since a monotonical increase in value with an increase in row and/or column number is preserved, the starting element chosen in this way will surely be the optimal one for the start of the search. As with mDTW, finding the optimal path is ensured by the fact that elements in each row, with the exception of the first one, inherently point to the optimal preceding element, since other candidates surrounding it will have higher values.

To enable easier implementation and reduce the computational time and effort, division of reference signal sample $b(j)$ with its corresponding lever length $l_b(j)$ does not have to be performed in each cycle of the comparison. Instead, all samples from the reference sequence can be divided with their corresponding lever lengths by piecewise division prior to the start of computation of the matrix d. This alteration has a considerable effect on the speed and number of calculations, since the reference sequence is significantly longer than the measurement vector. Similarly, since measurement vector Mv has first-in-first-out logic, at each comparison cycle, calculation of $a(i)$ and l_a(i) must be performed only for the newest sample m using $M_v\big(a(n), \Theta_v(n)\big)$, while other values can be reused.

4.5. Testing Results—Contact Tasks

The type of contact tasks addressed in this section are primarily so-called force contact tasks [23], in which both force and position need to be considered for successful execution. Here, they are further divided into two categories based on the direction of forces that are affecting the robot at the end effector side.

The first category includes tasks where the direction or profile of the external force is consistent in the robot‘s base frame, and it does not change regardless of the configuration of the robot. The most obvious example and the one that has the biggest impact on the quality of anomaly detection due to its frequency of application is the weight during load manipulation tasks. Other examples or variations of load manipulation include operations such as tending the press for bending, stamping, or clinching operations, in which the workpiece needs to be held throughout the process.

In the second group of tasks, the direction of external force is relative to the orientation of the end effector, i.e., considered consistent in the tool coordinate frame. These assignments mostly include processing and assembly tasks where the robot or its end effector are used to exert some force to the external object. Examples from this group include tasks such as drilling, screwdriving, riveting, nailing, stapling, snap-fitting, and bolting.

It is important to note that there are tasks that are a combination of the mentioned types. These tasks, however, are not the topic which will further be elaborated, since, in most cases, they can be separated into more basic operations to which the classification and related conclusions can be applied.

For testing purposes, all presented data were recorded using Denso VP-6242 six-axis industrial robot while it was performing its tasks, as shown in Figure 43. Testing of algorithms was done offline on a PC but with real measurement data from the robot, recorded using data exchange between the PC and the robot’s controller. In a realistic scenario, data gathering and processing would be implemented in a similar way, and therefore, the shown testing results are considered to closely resemble the results that would be obtained with hardware-in-loop testing. However, it is important to mention that data exchange and processing can be implemented on other platforms to achieve better performance or convenience [54].

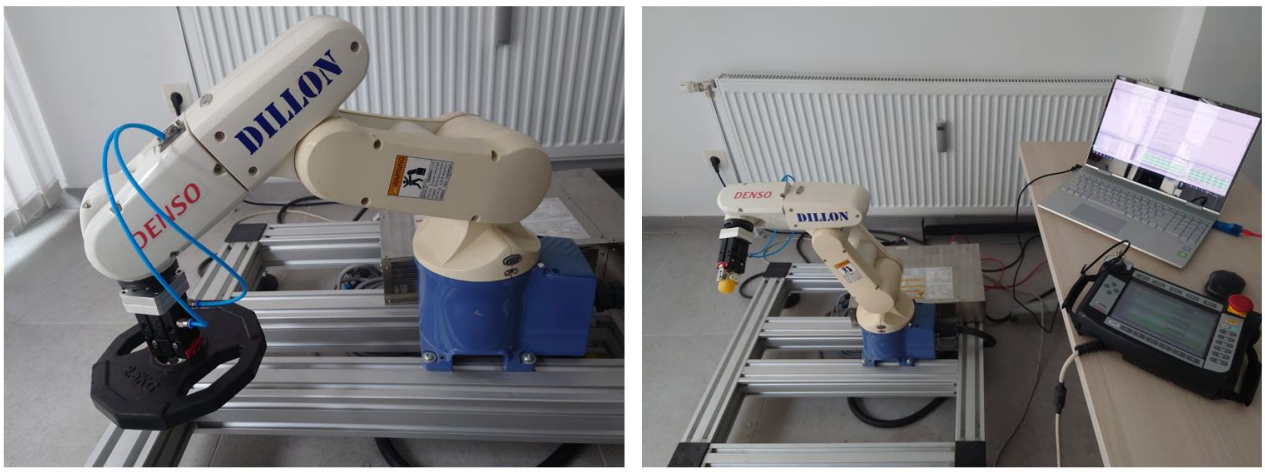

Figure 43. Measurement setups [59]. (left) Load manipulation contact task. (right) Snap-fit assembly contact task.

4.5.1. Consistent Base Frame Direction of Force

Following the general description of tasks within this category and relevant case of lever length calculations, this section aims to illustrate the inclusion and interpretation of contact task forces with consistent direction in the robot's base frame while performing a representative task example. Load manipulation is one of the typical tasks from this category and perhaps the most intuitive to understand since gravity is universally familiar.

To emulate possession of the optimal robot model, the first sequence of the movement was made without the presence of the load. The measurement from this movement represents an optimal model of the robot in which load dynamics are not included. The second sequence was recorded with actual load in a task where the robot simulates manipulation, in which the picking position is constant while placing positions differ one from the other. The first four placing positions feature identical orientations of the tool but with different distances from the base of the robot. The last two placing positions have identical tool orientations between them but differ compared to the first four and differ from each other in distances from the robot's base. Absolute values of joint currents were used in the manipulation contact task, together with joint angles, to complete the measurement vector.

Second axis analysis: The analysis of the interaction effects of the manipulation task is best illustrated with the example of the second axis, since it is affected by the weight of the load due to its orientation. Figure 44 shows one complete sequence of six picking and six placing operations. The upper part shows signals that correspond to the sequence recorded without load, shown in blue, a sequence recorded with actual load, shown in red, and the difference between them, shown in yellow. The difference between the first two signals is the actual deviation caused by the weight of the load, i.e., deviation caused by the contact task force, which was not included in the nominal dynamic model of the robot. The lower part of Figure 44 shows joint angles of the observed second axis, shown in blue, which are used to calculate lever lengths shown in red colour. When lever lengths are observed in parallel with deviations caused by the weight of the load, their effect on the profile of the deviation is evident. Deviation signal along with the joint angles form the measurement.

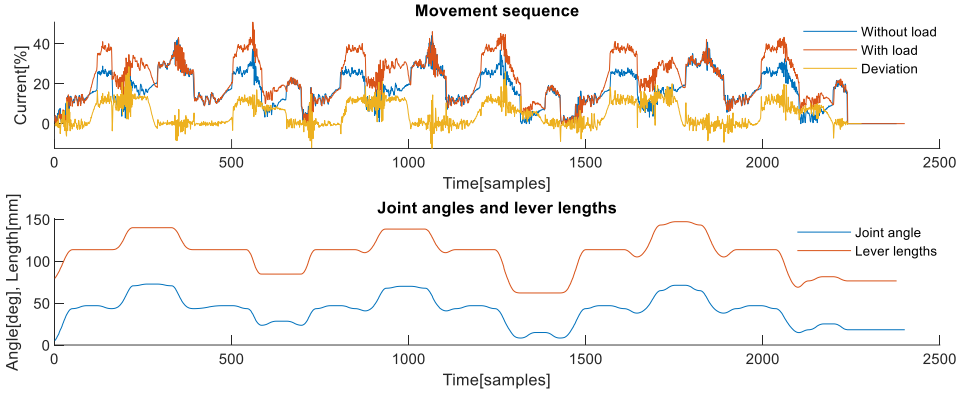

Figure 44. Signals obtained during the manipulation task from the 2nd axis [59]. (upper) Measurements of joint currents while the robot was performing the task without load and with the load. The difference between the two signals represented corresponds to the deviation profile originating from unmodelled dynamics of the contact task. Samples of deviation signal from each axis are one of the inputs to the proposed algorithm. (lower) Measurement of the joint angle is one of the inputs to the proposed algorithm. The algorithm calculates equivalent lever lengths in each sample based on DH parameters and joint angles from all axes.

The upper left part of Figure 45 shows the two sections of the deviations recorded on the second axis, which were used as examples of reference sequence and measurement vector for testing the algorithm's effectiveness. Both the reference sequence, shown in red, and the measurement vector, shown in yellow, were chosen due to their differences in shape caused primarily by the different direction and intensity of change in their corresponding lever lengths, shown in corresponding colours on the upper right part of Figure 45. One important aspect when

choosing the section that represents the reference sequence is that it needs to correspond to all phases that a contact task may have when the lever length is different from zero. In the case of manipulation contact tasks, it must include phases prior to picking up the load, lifting, transporting, lowering, releasing, and a brief section after the load is released. The lower left graph shows that the signals have different scales and that there is no obvious section of the reference sequence, shown in blue, that could reasonably be matched with the measurement vector shown in red. The lower right section of the same figure illustrates that after scaling, it is evident that, aside from the shift in time, there is a section of the scaled reference sequence that is similar in shape and scale to the scaled measurement vector.

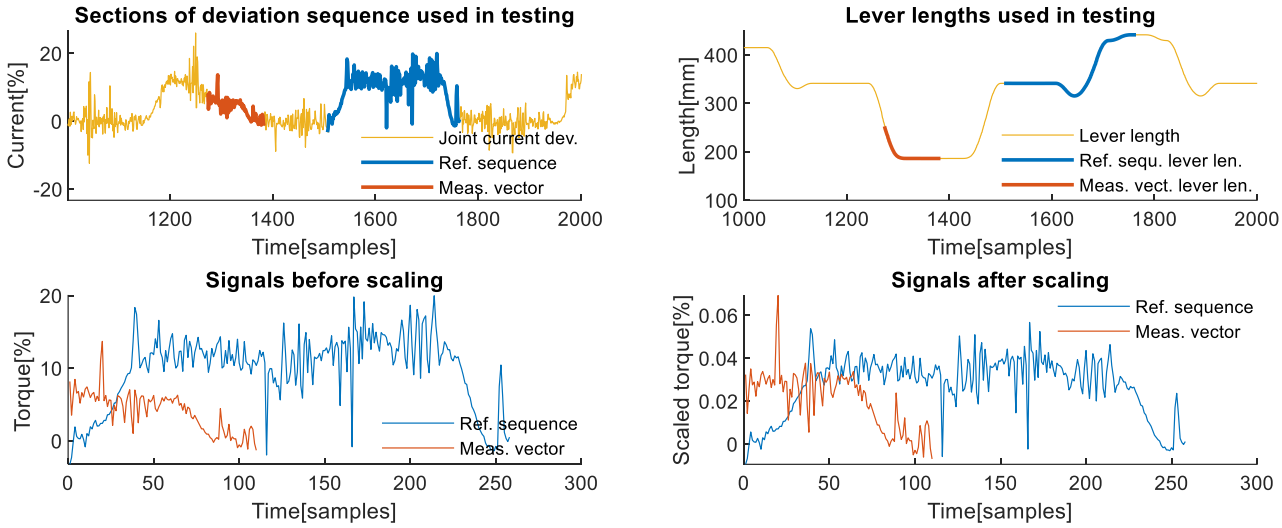

Figure 45. Signals used in the analysis for the 2nd axis [59]. (upper left) Sections of the deviation signal are used as a reference sequence and as a measurement vector during analysis. (upper right) Highlighted sections of the lever lengths corresponds to the chosen reference sequence and chosen measurement vector. (lower left) Original values of signals used as reference sequence and measurement vector shown prior to scaling by the algorithm. (lower right) Signals used as reference sequence and measurement vector after scaling was performed.

The upper left part of Figure 46 shows how the KA-mDTW algorithm has successfully matched these two signals, i.e., recognized the correct section of the reference sequence that needs to be matched with the measurement vector, as shown in the upper right part of Figure 46, enabling correct interpretation of the physical process, i.e., contact task dynamics. The corresponding matching cumulative error when KA-mDTW was used is 0.4173. For comparison purposes, the diagram shown on the lower left part of Figure 46 shows how the mDTW algorithm would match the two signals if they were in their original form. As shown, the entire measurement vector would be matched with the very end of the reference sequence, corresponding to the load-releasing phase. This matching completely misses the phase when the robot was stationary just prior to releasing the load, thus not interpreting the task dynamics correctly. Due to the incorrect selection of the corresponding section of the reference sequence, as seen in the lower middle part of Figure 46, the cumulative matching error for this case when mDTW was used is deceivingly small, at 0.2199. A correct way to calculate the real error would be to use the classical DTW algorithm to optimally match the section of the reference sequence highlighted on the upper right part of Figure 45, which was chosen by the KA-mDTW algorithm, with measurement vector, without applying scaling by the lever lengths. In that case, the cumulative error of the matching shown on the lower right part of Figure 46, accounting for the different scales of signals, is 3.1506, which is drastically higher than that of KA-mDTW.

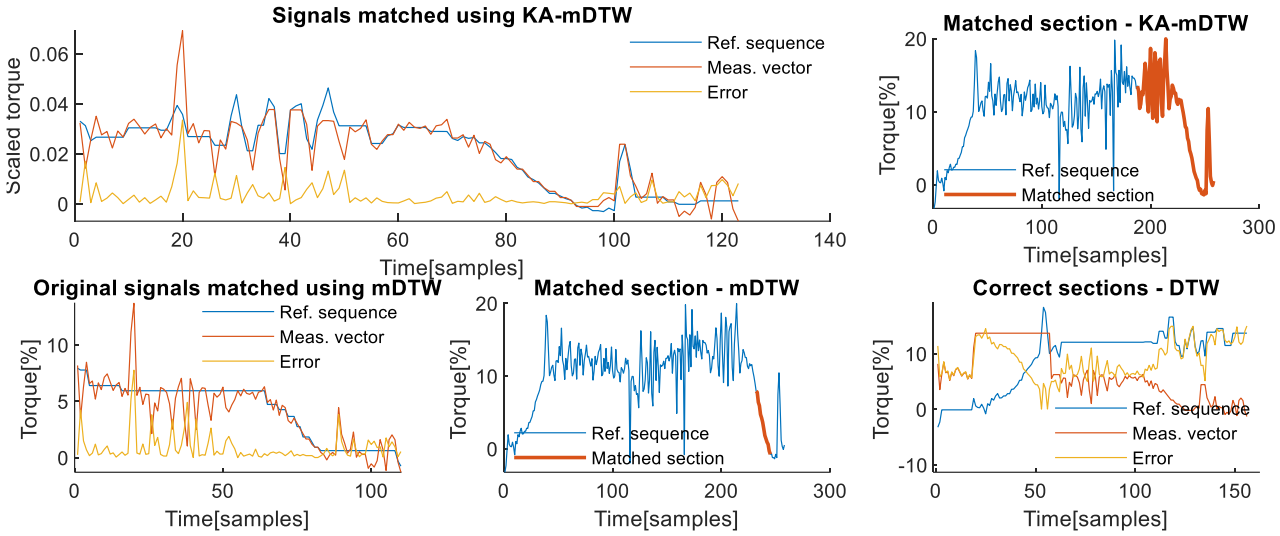

Figure 46. Matching of chosen signals from the 2nd axis [59]. (upper left) Optimally matched section of the reference sequence and the entire measurement vector and the sample-wise matching error. (upper right) Reference sequence profile with the highlighted section, which was correctly interpreted and optimally matched with the measurement vector. (lower left) Section of reference sequence incorrectly recognized and matched with measurement vector using mDTW without scaling the signals. (lower middle) Reference sequence profile with the highlighted section, which was incorrectly recognized and matched with the measurement vector using mDTW. (lower right) The section of the reference sequence identified by KA-mDTW is matched with the measurement vector using DTW algorithm to determine a realistic matching error.

Fourth axis analysis: The analysis of the manipulation contact task on the fourth axis enables illustrating another aspect related to the proper setting of the reference sequence. This axis was unaffected by the manipulation of load during the first four pick and place movements due to zero equivalent lever length, but it is affected during the fifth and sixth movements due to changes in tool orientation, as shown in Figure 47 in the upper sections. The reference sequence in blue must be chosen to include exclusively non-zero lever lengths for the observed joint if there is any possibility that that joint will be affected by the contact force. Otherwise, the joint cannot be used for effective comparison with the measurement vector. The middle right part of Figure 47 shows that the scaled reference has some significant peaks at the beginning, which are caused by division with small values of equivalent lever length. If the lever length is equal to zero, the algorithm does not work properly, which is a property that needs to be addressed. In the case of the second axis, it was explained that the reference signal must include all phases of the contact task that may occur when there is a lever. If only certain phases of the contact task are performed while the lever length is different from zero, then the reference signal does not have to include the phases during which the lever length is equal to zero, only those during which it is over a certain small threshold. If there is a theoretical chance that a certain phase not included in the reference signal may occur, then the reference sequence is not representative enough and needs to be changed. In case there are mutually exclusive non-zero lever length phases, the reference sequence can be composed of two separately recorded reference sequences that combined include all possible non-zero lever phases. In the presented scenario, only transporting, lowering, and the releasing phase of the manipulation task were possible due to the robot's configuration.

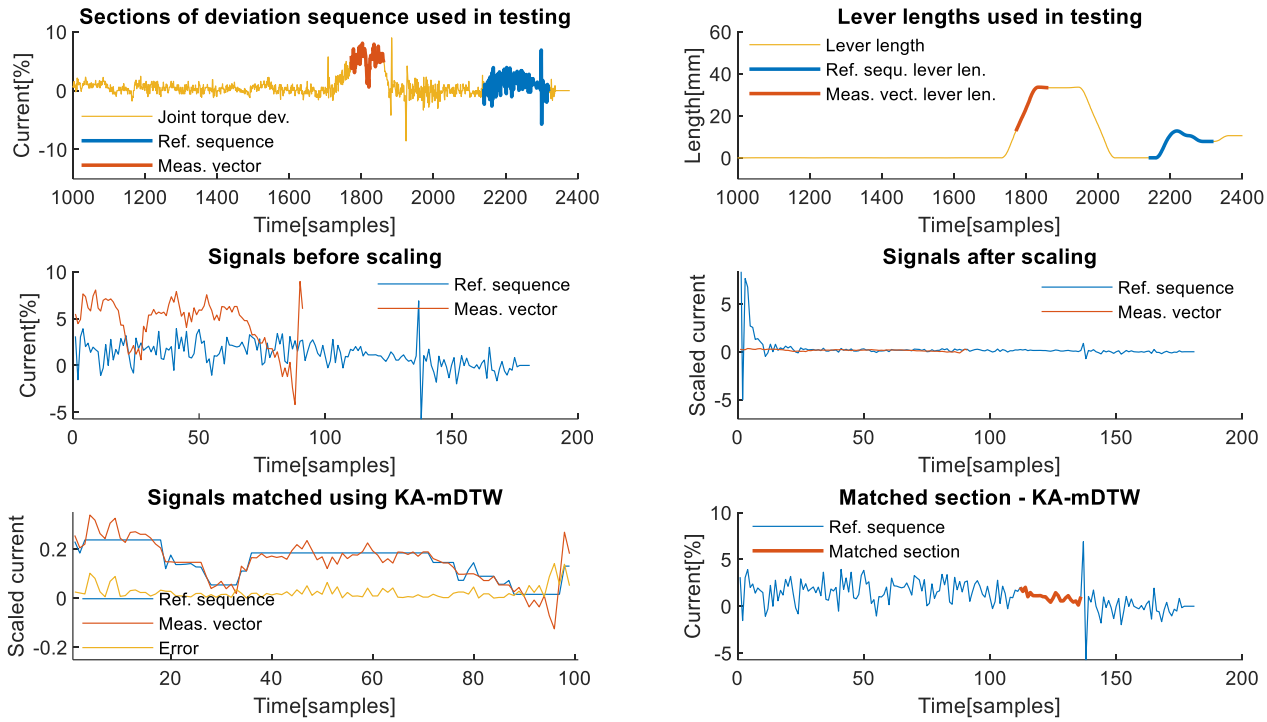

Figure 47. Analysis of example from the 4th axis [59]. (upper left) Deviation signal throughout the task with sections of the deviation signal used as reference sequence and the measurement vector. (upper right) Sections of the lever lengths corresponding to the chosen reference sequence and the chosen measurement vector. (middle left) Signal used as reference sequence and measurement vector in their original values prior to scaling by the algorithm. (middle right) Signals used as reference sequence and measurement vector after scaling was applied. (lower left) Graph shows optimally matched section of the reference sequence and the entire measurement vector as well as the sample-wise matching error. (lower right) Reference sequence profile with the highlighted section, which was correctly interpreted and optimally matched with the measurement vector.

The lower left part of Figure 47 shows that the KA-mDTW algorithm can optimally match the reference sequence shown in blue, and the measurement vector shown in red with a cumulative error is 2.3941. The lower right part highlights in red the section of the reference to which the measurement vector was matched, which corresponds to the actual situation, although it is slightly shorter. For the same signals without scaling, the cumulative sum of error, compensated for the different scaling, is 19.2877, even with matching with the incorrect section of the reference subsequence. When DTW is applied to the correct section of the reference sequence, the realistic cumulative error is even higher, at 28.2313, showing the importance of the scaling phase of the KA-mDTW algorithm.

4.5.2. Consistent Tool Frame Direction of Force

As a representative for the type of contact forces, a snap-fit assembly task was chosen. It features a profile that can easily be mistaken for a collision due to its nature [56]. Therefore, the importance of proper interpretation of this signal is very high, as it may have serious consequences for production. The effects that this assembly contact task dynamics has on joint manifest themselves in the form of peaks in measurement values. With an aim to demonstrate the versatility of the algorithm in terms of signal availability using this example, instead of absolute values of currents, the current-based estimations of torque were used together with measurements of joint angles. The values of this signal were readily available as outputs from the Denso VP-6242 robot, and in this section, they will interchangeably be referred to as torques or torque estimates.

First axis analysis: The first axis is interesting for analysis, because it is usually not affected by gravity and is theoretically relatively insensitive to the influence of external forces. The latter is due to gearing ratios and powerful motors, which need to be used because of the high moments of inertia that it must withstand. The analysis of the effects that the dynamics of the snap-fit assembly contact task have on the first axis of the robot was performed using measurements and values presented in Figure 48. The deviation signal, shown in yellow on the left graph of Figure 48, which is used as input to KA-mDTW algorithm together with joint angles shown in blue on the right part of the same figure, has different dynamics compared to the corresponding signals analysed in the manipulation task examples. The reason is that during this task, the robot only moved to assume new positions for the assembly, while the force was exerted on the work object while the robot was stationary. This situation is present in various contact tasks of this type, and therefore, the conclusions made during this analysis are not limited to snap-fit assembly.

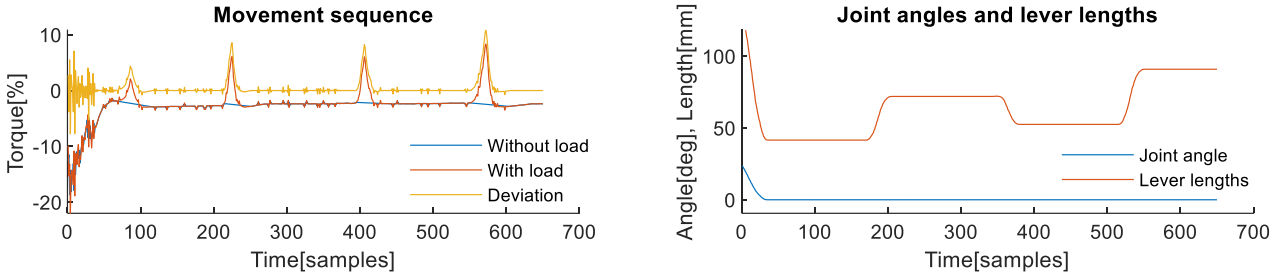

Figure 48. Signals obtained during snap-fit assembly task from the 1st axis [59]. (left) Measurements of joint torques while the robot was performing the task without and with assembly of the object. The difference between them corresponds to the profile of the contact-task-induced deviation. (right) Measurement of the joint angle used to calculate equivalent lever lengths.

Similar to previous examples, Figure 49 in the upper left part shows sections of the deviation sequence, shown in yellow, which were used as reference sequence and measurement vector, highlighted in blue and red, respectively. The same is valid for lever lengths on the upper right graph of the same figure. From the lower left part of Figure 49, it is possible to observe that the selected reference sequence shown in blue and the measurement vector shown in red are similar only in shape but not in scale and values, although they originate from the same task. Scaling, which is an integral part of the KA-mDTW algorithm, makes these signals much easier to compare and interpret, as shown in the lower right part of Figure 50.

Figure 50 in the upper left part shows that the observed signals shown in blue and red were correctly matched, which is confirmed by the error signal shown in the yellow and cumulative matching error of 0.0207. The reference sequence segment corresponding to the measurement vector was correctly identified, as highlighted on the upper right part of the same figure, indicating that the contact task dynamics would be correctly interpreted. The most striking observation is that the mDTW algorithm is powerless in this situation and only manages to match the entire measurement with a single point from the reference sequence, which can be seen in the lower left and middle parts of Figure 50. Although striking, this result from mDTW is expected, and it is truly the best result it could theoretically achieve. Nevertheless, the error shown in yellow on the lower left graph has a cumulative value of 0.3106, which is significantly higher than with KA-mDTW. However, the biggest implication of this matching is that the contact force would be completely wrongly interpreted and would surely indicate some erroneous state. When the DTW algorithm is used to match the section of the reference sequence identified by the KA-mDTW with the measurement vector, the best result it can achieve is 0.7009.

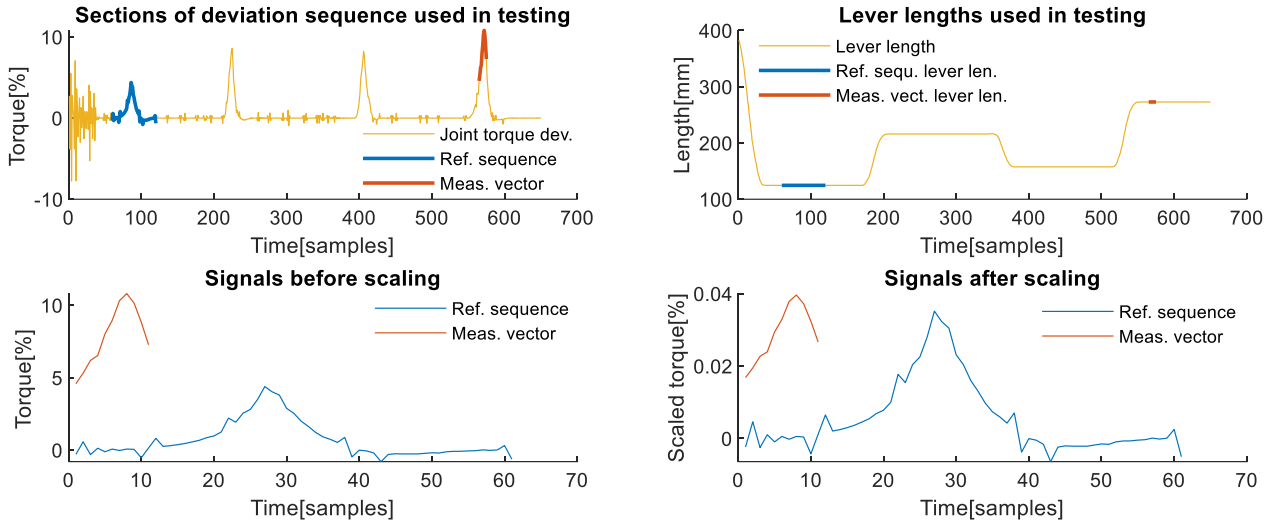

Figure 49. Matching chosen signals from the 1st axis [59]. (upper left) Sections of the deviation signal used as reference sequence and the measurement vector during analysis. (upper right) Sections of the lever lengths corresponding to chosen testing signals. (lower left) Signals used as reference sequence and measurement vector in their original values prior to scaling by the algorithm. (lower right) Signals used as reference sequence and measurement vector after scaling was applied.

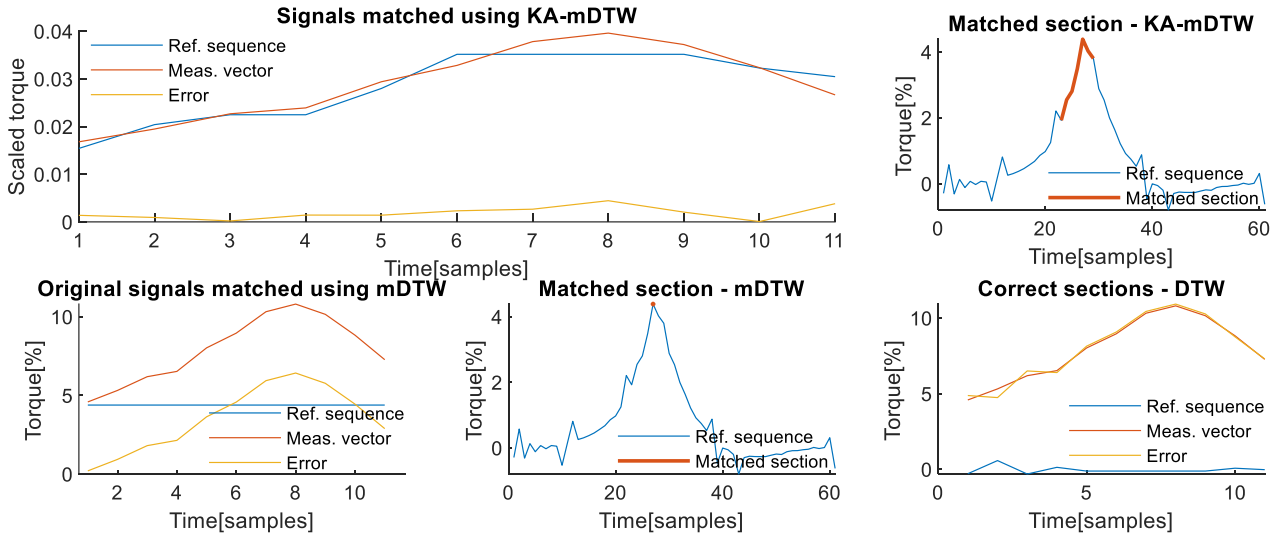

Figure 50. Matching of chosen signals from the 1st axis [59]. (upper left) Graph shows optimally matched section of the reference sequence and the entire measurement vector as well as the sample-wise matching error. (upper right) Reference sequence profile with the highlighted section, which was optimally matched with the measurement vector. The highlighted section correctly interprets the phase of the contact task contained in the measurement vector. (lower left) Section of reference sequence matched with measurement vector using mDTW without scaling the signals. A matching error has the same shape as the measurement vector, since the entire vector was only matched with a single point of reference, which was closest in value. (lower middle) Reference sequence profile with the highlighted section, which is just a point matched with the measurement vector using mDTW. The highlighted section corresponds to the peak of the reference signal, since it was the closest to all the values within the measurement vector. (lower right) The section of the reference sequence, which was optimally matched using KA-mDTW from the upper right graph, is matched with the measurement vector using DTW algorithm to determine a realistic matching error.

Third axis analysis: The third axis of the robot is one of the most affected axes, regardless of the contact task type. The selected example aims to illustrate two things, the first of which can be observed from the upper graphs of Figure 51. In the upper left graph, the deviation signal shown in yellow on this axis shows that the first and the third deviation caused by the snap-fit contact task have similar intensity due to the identical corresponding equivalent lever lengths, which is also valid for the second and fourth deviation-induced peak.

The signals chosen for comparison as reference sequence and measurement vector, highlighted in blue and red, respectively, on the upper parts of Figure 51, demonstrate performance in the situation opposite of the one analysed for the first robot axis. In a situation when the reference signal, shown in blue on the middle right part of Figure 51, has higher values than the measurement vector shown in red, the performance of KA-mDTW algorithm is unaltered. As shown in the middle right part of the same figure, it correctly identifies the section of the reference sequence and optimally matches it with a cumulative matching error of 0.0518. The mDTW algorithm performs better than in the example from the first axis with a cumulative matching error of 0.0473, which is due to the incorrect section with which it was matched, leading to dynamics misinterpretation. The equivalent matching error calculated using DTW is 0.1640.

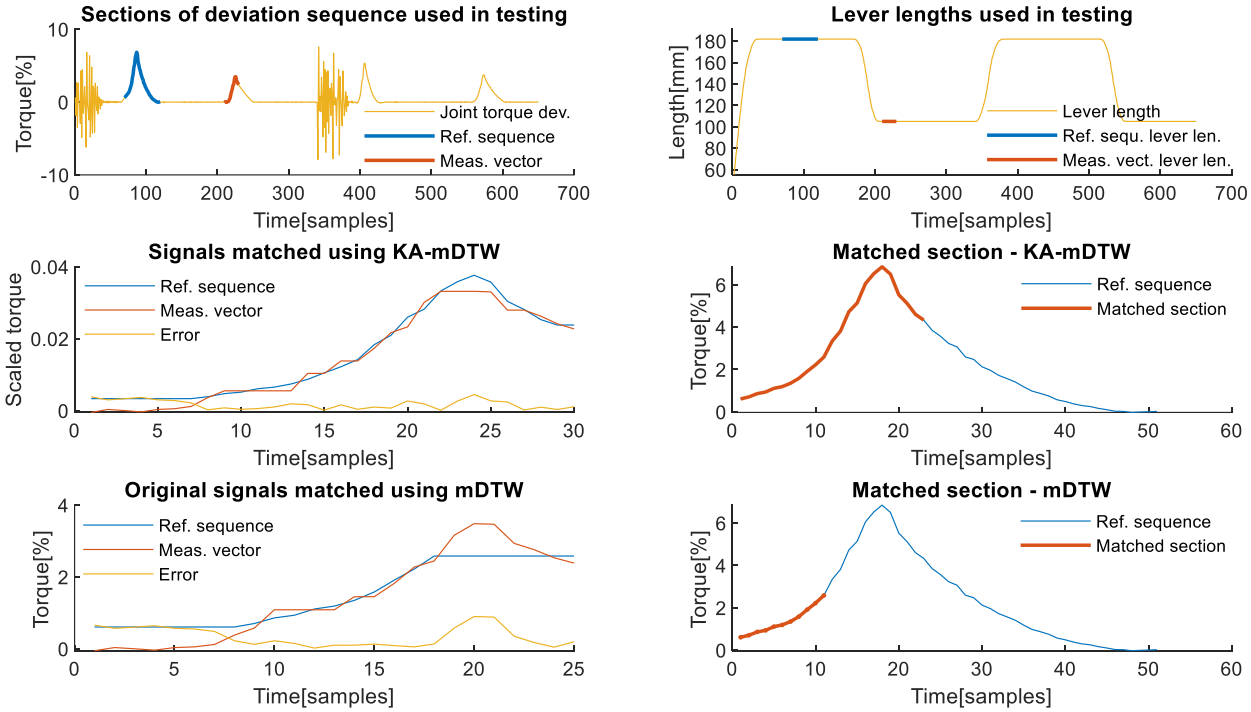

Figure 51. Matching chosen signals from the 3rd axis [59].(upper left) Section of the deviation signal used as a reference sequence during analysis, and the section used as the measurement vector. (upper right) Sections of the lever lengths corresponding to the chosen reference sequence and the chosen measurement vector. (middle left) Optimally matched section of the reference sequence, the entire measurement vector, and the sample-wise matching error. (middle right) Reference sequence profile with the highlighted section, which was correctly recognized, optimally matched with the measurement vector. (lower left) Section of reference sequence matched with measurement vector using mDTW without scaling the signals and matching error. (lower right).

4.6. Chapter discussions

Contents of this Chapter address the issue of unmodelled dynamics related to contact tasks performed by industrial robots. The reasons for not considering contact task dynamics include modelling or implementation complexity, insufficient knowledge of the task, and underestimation of the effect they have. Consequences, however, are almost universally negative with regards to the reliability, observability, and predictability of the process involving them. From the perspectives of Industry 4.0 and 5.0, where collision erroneous state detection and handling, as well as operation context brokerage, are the basis of successful implementation, each of the mentioned drawbacks gains more significance.

This Chapter proposes to include the contact task dynamics implicitly by observing their effects on individual robot joints and understanding how this influence transforms with changes in the spatial relation of joints. Correct interpretation of the effects of contact forces enables the identification of phases of the task, and their comparison with expected values enables the detection of erroneous states.

The underlying idea proposes that the effects of contact task dynamics are observed through deviations of joint currents or torques from the nominal values. Samples from a representative task execution form a reference sequence with which all future samples of deviation will be compared. Since the effects that an external contact force may have on a robot joint change with changes in the spatial relation between the joints, so do the profiles of the detected deviation.

To enable comparison between different profiles of deviations, an equivalent lever length parameter is introduced based on the considerations of its correlation with the deviation signal. The equivalent lever length is calculated for each joint at each sample based on the DH parameters of the robot and measurements of the joint angles and used to scale the corresponding sample of joint current or torque. The sample-wise scaling of samples of the reference sequence and the measurement vector enables their reliable comparison using a modification of Dynamic Time Warping (mDTW), a previously developed algorithm for real-time matching of signals with content and sampling differences.

Fusion of equivalent lever lengths scaling and mDTW resulted in the newly proposed Kinematically Augmented mDTW (KA-mDTW). This versatile and efficient algorithm introduces robot kinematics into flexible time series matching. Furthermore, the proposed changes influence the rules for calculation of the d matrix of mDTW algorithm, based on which the matching surface is shaped and optimally matched pairs are determined.

The KA-mDTW algorithm was thoroughly tested using real measurements from an industrial robot performing load manipulation and snap-fit assembly tasks as representative examples from two groups of contact tasks. For each example, the efficiency of the algorithm was demonstrated and commented on with regards to the cumulative matching error compared to the mDTW algorithm, as well as the correct identification of the matched section of the reference sequence. Due to its inherent signal scaling ability, KA-mDTW's cumulative matching error was shown to be drastically lower than the equivalent error of mDTW. Correctly matched sections of the reference sequence enable proper interpretation of the phases of the contact task dynamics, as well as reliable detections of erroneous states.

The testing phase also commented that the reference sequence must be representative of the contact task, including all possible effects that the task may have on the joint. These include all phases of the task during which the equivalent lever length is non-zero. That effectively means that the algorithm can match detected deviation successfully only if the reference sequence includes an example of the same phase of the task that caused the deviation. It was also noted that references recorded with small lever lengths cause peaks due to scaling with small values. For the same reason, the reference sequence must not include samples during which the equivalent lever length was zero.

The proposed approach and the algorithm successfully avoided implementation issues by relying on versatile automatic parameter identification and processing automation. However, the fact that the algorithm's performance depends on setting the proper reference sequence leaves room

for user-induced performance reduction. Further research will examine algorithms for evaluation of the reference sequence and information they include. Another direction will focus on predicting time intervals when the deviation is expected, i.e., the intentional contact of the robot and its surroundings. To this end, many approaches are considered, including Artificial Intelligence and some other methods of Soft-computing as the most promising [75].

From the standpoint of implementation in scope of a hybrid approach, the KA-mDTW algorithm solves field of application issues related to the m-DTW-based algorithm presented in the previous Chapter. Once the deviation signal is generated by the model-based part, the KA-mDTW part of the algorithm can effectively incorporate the unmodelled dynamics of the robot motion or the contact task. Thanks to the inclusion of the kinematic parameters, such implementation can effectively replace the commonly used force/torque sensors while offering a more reliable reaction to external forces, intentional and unintentional alike. Most importantly, such implementation would also enable inclusion of task dynamics not reflected on the end-effector, but also detection of collisions that may occur along the kinematic chain between the robot base and the flange.

5. GENERAL CONCLUSION

The work presented in this thesis aimed to enable better understanding and safer interaction of industrial robots with their surroundings. To that end, several approaches were developed and presented to provide means for correct interaction interpretation based on readily available sensors and measurements found in industrial robots with closed control architecture.

The common approach in all presented algorithms for detection of interactions was to observe, analyse and interpret how the influence of interaction forces manifests itself on the measurements commonly available on industrial robots. To this end, available signals were first analysed to draw conclusions about their properties. The observations related to their dynamics, repeatability and sampling were used to design different non-model-based interaction detection algorithms.

The principle on which the interaction detection is based in all presented algorithms takes advantage of the repetitive nature of majority of robot tasks. The basic idea is that the reference sequence recorded during one exemplary cycle of execution is compared with measurements from all future cycles of the task execution to determine the presence and type of interactions.

The first two presented algorithms were intended primarily for the detection of the unintentional interactions, i.e., collisions. The first presented algorithm, developed for integration on the robot controller itself, relied on practical solutions for simple yet reliable collision detection and served as a good basis for understanding the fields in which improvements can be made.

Pursue for improvements led the to the shift of attention on implementation on a PC, which enabled development of a second, more complex and performant collision detection algorithm. The key contribution and enabling factor for overcoming sampling issues was the development of modified Dynamic Time Warping (mDTW) method, thanks to which the principles of time warping were enabled matching sequences with different content in real time. The mDTW enabled implementation of more sensitive and comprehensive detection rules based on statistical properties of the signal. The algorithm's potential for detection of intentional interactions in form of a hybrid approach was considered and discussed with regards to the properties of the deviations induced by the typical examples of contact tasks. The overall positive evaluation of such prospect and the m-DTW based detection in general was found to be limited only to the repetitive tasks which involve identical movement of the robot arm.

The third presented detection algorithm was developed to extend the application domain of mDTW-based detection by introduction of kinematics-related parameters which would enable its implementation in tasks with similar type of movement. To this end, two algorithms for automatic identification of kinematics parameters were developed. The first algorithm identifies parameters of robot arm or arbitrary configuration based on partial-pose measurements of a point of interest mounted on the robot's end effector. Based on only three position measurements per robot joint, such algorithm can calculate the complete kinematic model in DH notation. The second algorithm for kinematic parameter identification was developed as a robot tool calibration unit capable of automatic calculation of robot's end effector parameters. Based on analysis of images from two orthogonal planes, this solution enables not only identification of the end effector parameters, but also their verification and recovery in case of detected collision.

The two algorithms for identification of kinematic parameters of the robot and its end-effector enable the further evolution of dynamic warping capabilities in form of Kinematic Augmented mDTW (KA-mDTW) method. This crucially impactful improvement enables detection of interactions for movements with different spatial relations of joints than those featured during recording of the reference sequence. Consequently, with properly chosen references, this improvement has eliminated the limitations for robot's motion within the same type of the task, and by doing so, has also overcome the biggest disadvantage of this type of interaction detection, as well as for the hybrid approach mentioned earlier.

All results and conclusions were validated through experimental testing on two brands of industrial robots to further demonstrate their platform-indifference. Special intention was given to the design of procedures and algorithms which were developed with consideration to the constrictions imposed by the physical design of the robot and its typical operating environment. To reduce the potential for operator-induced errors and facilitate integration into the production environment.

The impact of the presented work can best be observed through the prism of Industry 4.0 and advanced agile robotization. In event-driven and highly interconnected production environments the understanding of operation context of production resources is of outmost importance for the safety and production optimization. Understanding the operation context of the robot as one of the crucial enablers in such environments is the necessary step in this direction, especially in scope of proper execution of contact tasks which are and will remain by far the most predominant type of applications of industrial robots.

REFERENCES

[1] N. Pedrocchi, F. Vicentini, M. Matteo i L. M. Tosatti, „Safe Human-Robot Cooperation in an Industrial Environment," *International Journal of Advanced Robotic Systems,* t. 10, p. 27, 2013.

[2] Z. Gordić i K. Jovanović, „Influence of Unmodelled External Forces on the Quality of Collision Detection," u *Advances in Service and Industrial Robotics,* Cham, 2020.

[3] S. Haddadin, A. Albu-Schäeffer, A. Luca i G. Hirzinger, „Evaluation of Collision Detection and Reaction for a Human-Friendly Robot on Biological Tissues," 2008.

[4] S. Haddadin, A. Albu-Schäffer i G. Hirzinger, „Safe Physical Human-Robot Interaction: Measurements, Analysis and New Insights," u *Robotics Research,* Berlin, 2011.

[5] S. Haddadin, A. Albu-Schäeffer i G. Hirzinger, „Safety Evaluation of Physical Human-Robot Interaction via Crash-Testing," 2007.

[6] International Federation of Robotics, „World Robotics 2021—Industrial Robots," International Federation of Robotics, Frenkfurt, Germany, 2021.

[7] A. Perzylo, M. Rickert, B. Kahl, N. Somani, C. Lehmann, A. Kuss, S. Profanter, A. B. Beck, M. Haage, M. Rath Hansen, M. T. Nibe, M. A. Roa, O. Sörnmo, S. Gestegård Robertz, U. Thomas, G. Veiga, E. A. Topp, I. Kessler i M. Danzer, „SMErobotics: Smart Robots for Flexible Manufacturing," *IEEE Robotics & Automation Magazine,* t. 26, pp. 78-90, 2019.

[8] A. De Luca i F. Flacco, „Integrated control for pHRI: Collision avoidance, detection, reaction and collaboration," u *2012 4th IEEE RAS & EMBS International Conference on Biomedical Robotics and Biomechatronics (BioRob),* 2012.

[9] F. Flacco i A. De Luca, „Safe physical human-robot collaboration," u *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems,* 2013.

[10] A. de Luca i R. Mattone, „Sensorless Robot Collision Detection and Hybrid Force/Motion Control," u *Proceedings of the 2005 IEEE International Conference on Robotics and Automation,* 2005.

[11] E. Mariotti, E. Magrini i A. D. Luca, „Admittance Control for Human-Robot Interaction Using an Industrial Robot Equipped with a F/T Sensor," u *2019 International Conference on Robotics and Automation (ICRA),* 2019.

[12] M. Geravand, F. Flacco i A. De Luca, „Human-robot physical interaction and collaboration using an industrial robot with a closed control architecture," u *2013 IEEE International Conference on Robotics and Automation,* 2013.

[13] Z. Gordić i K. Jovanović, „Fully Integrated Torque-Based Collision Detection in Periodic Tasks for Industrial Robots with Closed Control Architecture," u *Advances in Service and Industrial Robotics,* Cham, 2019.

[14] Z. Gordic i K. Jovanovic, „Collision Detection on Industrial Robots in Repetitive Tasks Using Modified Dynamic Time Warping," *Robotica,* pp. 1-20, 2019.

[15] M. Makarov, A. Caldas, M. Grossard, P. Rodrıguez-Ayerbe i D. Dumur, „Adaptive Filtering for Robust Proprioceptive Robot Impact Detection Under Model Uncertainties," *IEEE/ASME Transactions on Mechatronics,* t. 19, br. 6, pp. 1917-1928, 2014.

[16] A. De Luca, A. Albu-Schaffer, S. Haddadin i G. Hirzinger, „Collision Detection and Safe Reaction with the DLR-III Lightweight Manipulator Arm," u *2006 IEEE/RSJ International Conference on Intelligent Robots and Systems,* 2006.

[17] L. D. Avendano-Valencia, F. Dimeas i N. Aspragathos, „Human - Robot collision detection and identification based on fuzzy and time series modelling," *Robotica,* t. 33, May 2014.

[18] P. Pastor, L. Righetti, M. Kalakrishnan i S. Schaal, „Online movement adaptation based on previous sensor experiences," u *2011 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2011.

[19] S. Haddadin, A. Albu-Schaffer, A. De Luca i G. Hirzinger, „Collision Detection and Reaction: A Contribution to Safe Physical Human-Robot Interaction," u *2008 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2008.

[20] Y. Yamada, Y. Hirasawa, S. Y. Huang i Y. Umetani, „Fail-safe human/robot contact in the safety space," u *Proceedings 5th IEEE International Workshop on Robot and Human Communication. RO-MAN'96 TSUKUBA*, 1996.

[21] M. S. Erden i T. Tomiyama, „Human-Intent Detection and Physically Interactive Control of a Robot Without Force Sensors," *IEEE Transactions on Robotics*, t. 26, pp. 370-382, 2010.

[22] P. Cao, Y. Gan i X. Dai, „Model-based sensorless robot collision detection under model uncertainties with a fast dynamics identification," *International Journal of Advanced Robotic Systems*, t. 16, p. 1729881419853713, 2019.

[23] M. Vukobratović i V. Potkonjak, „Dynamics of Robots with Contact Tasks. Intelligent Systems," u *Control and Automation: Science and Engineering Mechanism and Machine Theory*, Springer: Dordrecht, 2003.

[24] S. Hagane, L. Rincon Ardila, T. Katsumata, V. Bonnet, P. Fraisse i G. Venture, „Adaptive Generalized Predictive Controller and Cartesian Force Control for Robot Arm Using Dynamics and Geometric Identification," *Journal of Robotics and Mechatronics*, t. 30, pp. 927-942, December 2018.

[25] W. Khalil i F. Bennis, „Symbolic Calculation of the Base Inertial Parameters of Closed-Loop Robots," *The International Journal of Robotics Research*, t. 14, pp. 112-128, 1995.

[26] P. K. Khosla i T. Kanade, „Parameter identification of robot dynamics," u *1985 24th IEEE Conference on Decision and Control*, 1985.

[27] M. Gautier, „Numerical calculation of the base inertial parameters of robots," u *Proceedings., IEEE International Conference on Robotics and Automation*, 1990.

[28] M. Gautier i W. Khalil, „Direct calculation of minimum set of inertial parameters of serial robots," *IEEE Transactions on Robotics and Automation*, t. 6, pp. 368-373, 1990.

[29] M. Gautier i W. Khalil, „On the identification of the inertial parameters of robots," u *Proceedings of the 27th IEEE Conference on Decision and Control*, 1988.

[30] A. Janot, P.-O. Vandanjon i M. Gautier, „A Generic Instrumental Variable Approach for Industrial Robot Identification," *IEEE Transactions on Control Systems Technology*, t. 22, pp. 132-145, 2014.

[31] M. Brunot, A. Janot, F. J. Carrillo i H. Garnier, „A Pragmatic and Systematic Statistical Analysis for Identification of Industrial Robots," 2017.

[32] L. Ding, H. Wu, Y. Yao i Y. Yang, „Dynamic Model Identification for 6-DOF Industrial Robots," *Journal of Robotics*, t. 2015, pp. 1-9, October 2015.

[33] K. Godfrey, „Identification of parametric models from experimental data [Book Review]," *Automatic Control, IEEE Transactions on*, t. 44, pp. 2321-2322, January 2000.

[34] V. Bargsten, P. Zometa i R. Findeisen, „Modeling, parameter identification and model-based control of a lightweight robotic manipulator," u *2013 IEEE International Conference on Control Applications (CCA)*, 2013.

[35] J. Jin i N. Gans, „Parameter identification for industrial robots with a fast and robust trajectory design approach," *Robotics and Computer-Integrated Manufacturing*, t. 31, pp. 21-29, 2015.

[36] A. Bahloul, S. Tliba i Y. Chitour, „Dynamic Parameters Identification of an Industrial Robot With and Without Payload," *IFAC-PapersOnLine*, t. 51, pp. 443-448, 2018.

[37] A.-N. Sharkawy, P. N. Koustoumpardis i N. A. Aspragathos, „Manipulator Collision Detection and Collided Link Identification Based on Neural Networks," u *Advances in Service and Industrial Robotics*, Cham, 2019.

[38] Z. Gordić i K. Jovanovic, „Collision Detection on Industrial Robots in Repetitive Tasks Using Modified Dynamic Time Warping," *Robotica,* t. 38, pp. 1-20, October 2019.

[39] C.-N. Cho, J.-H. Kim, Y.-L. Kim, J.-B. Song i J.-H. Kyung, „Collision Detection Algorithm to Distinguish Between Intended Contact and Unexpected Collision," *Advanced Robotics,* t. 26, pp. 1-16, November 2012.

[40] A. D. Santis, B. Siciliano, A. D. Luca i A. Bicchi, „An atlas of physical human–robot interaction," *Mechanism and Machine Theory,* t. 43, pp. 253-270, 2008.

[41] M. Zinn, O. Khatib, B. Roth i J. K. Salisbury, „Playing it safe [human-friendly robots]," *IEEE Robotics & Automation Magazine,* t. 11, pp. 12-21, 2004.

[42] N. Knežević, M. Majstorović, Z. Gordić, K. Jovanović i V. Potkonjak, „Hazard Identification, Risk Asse Assessment and Safety Integration for Flexible Roboti Cell," u *Proceedings of 4th International Conference on Electrical, Electronics and Computing Engineering, IcETRAN 2017,*, Kladovo, Serbia, 2017.

[43] R. J. Kirschner, A. Kurdas, K. Karacan, P. Junge, S. A. Baradaran Birjandi, N. Mansfeld, S. Abdolshah i S. Haddadin, „Towards a Reference Framework for Tactile Robot Performance and Safety Benchmarking," u *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2021.

[44] E. Magrini, F. Flacco i A. De Luca, „Control of generalized contact motion and force in physical human-robot interaction," u *2015 IEEE International Conference on Robotics and Automation (ICRA)*, 2015.

[45] J. Heinzmann i A. Zelinsky, „Quantitative Safety Guarantees for Physical Human-Robot Interaction," *The International Journal of Robotics Research,* t. 22, pp. 479-504, 2003.

[46] D. F. Silva, G. E. A. P. A. Batista i E. Keogh, „Prefix and Suffix Invariant Dynamic Time Warping," u *2016 IEEE 16th International Conference on Data Mining (ICDM)*, 2016.

[47] P. Tormene, T. Giorgino, S. Quaglini i M. Stefanelli, „Matching incomplete time series with dynamic time warping: an algorithm and an application to post-stroke rehabilitation," *Artificial Intelligence in Medicine,* t. 45, pp. 11-34, 2009.

[48] G. Navarro, „A Guided Tour to Approximate String Matching," *ACM Computing Surveys,* t. 33, April 2000.

[49] D. S. Hirschberg, „A linear space algorithm for computing maximal common subsequences," *Communications of the ACM*, t. 18, pp. 341-343, 1975.

[50] C. Jekel, G. Venter, M. Venter, N. Stander i R. Haftka, „Similarity measures for identifying material parameters from hysteresis loops using inverse analysis," *International Journal of Material Forming,* t. 12, May 2019.

[51] „Dynamic Time Warping," u *Information Retrieval for Music and Motion*, Berlin, Heidelberg: Springer Berlin Heidelberg, 2007, p. 69–84.

[52] I. Oregi, A. Pérez, J. D. Ser i J. A. Lozano, „On-line Elastic Similarity Measures for time series," *Pattern Recognition,* t. 88, pp. 506-517, 2019.

[53] I. Oregi, A. Pérez, J. Del Ser i J. A. Lozano, „On-Line Dynamic Time Warping for Streaming Time Series," u *Machine Learning and Knowledge Discovery in Databases*, Cham, 2017.

[54] D. Sart, A. Mueen, W. Najjar, E. Keogh i V. Niennattrakul, „Accelerating Dynamic Time Warping Subsequence Search with GPUs and FPGAs," u *2010 IEEE International Conference on Data Mining*, 2010.

[55] L. J. Latecki, V. Megalooikonomou, Q. Wang, R. Lakaemper, C. Ratanamahatana i E. Keogh, „Elastic Partial Matching of Time Series," 2005.

[56] Z. Gordić i K. Jovanović, „Influence of Unmodelled External Forces on the Quality of Collision Detection,“ u *Advances in Service and Industrial Robotics - Proceedings of the 28th*, Kaiserslautern, 2019.

[57] F. Ferraguti, C. Secchi i C. Fantuzzi, „A tank-based approach to impedance control with variable stiffness,“ u *2013 IEEE International Conference on Robotics and Automation*, 2013.

[58] C. Schindlbeck i S. Haddadin, „Unified passivity-based Cartesian force/impedance control for rigid and flexible joint robots via task-energy tanks,“ u *2015 IEEE International Conference on Robotics and Automation (ICRA)*, 2015.

[59] Z. Gordić i K. Jovanović, „A Framework for Inclusion of Unmodelled Contact Tasks Dynamics in Industrial Robotics,“ *Sensors*, t. 22, 2022.

[60] Z. Gordić i K. Jovanović, „Partial Pose Measurements for Identification of Denavit-Hartenberg Parameters of an Industrial Robot,“ u *Proceedings of the 4th International Conference on Electrical, Electronic and Computing Engineering (IcETRAN 2017)*, Kladovo, 2017.

[61] C. G. Rajeevlochana, S. Saha i S. Kumar, „Automatic Extraction of DH Parameters of Serial Manipulators using Line Geometry,“ 2012.

[62] H.-N. Nguyen, J. Zhou i H.-J. Kang, „A New Full Pose Measurement Method for Robot Calibration,“ *Sensors (Basel, Switzerland)*, t. 13, pp. 9132-47, July 2013.

[63] A. Reddy, „Difference Between Denavit - Hartenberg (D-H) Clasical and Modified Conventions for Forward Kinematics of Robots with Case Study,“ 2014.

[64] C. Faria, J. L. Vilaça, S. Monteiro, W. Erlhagen i E. Bicho, „Automatic Denavit-Hartenberg Parameter Identification for Serial Manipulators,“ u *IECON 2019 - 45th Annual Conference of the IEEE Industrial Electronics Society*, 2019.

[65] L. Žlajpah i T. Petrič, „Geometric Identification of Denavit-Hartenberg Parameters with Optical Measuring System,“ u *Advances in Service and Industrial Robotics*, Cham, 2022.

[66] B. Borovac, S. Đorđević, M. Rašić i M. Raković, Industrijska robotika, Novi Sad, 2016.

[67] H. Bruyninckx, Robot Kinematics and Dynamics, Leuven, Belgium: Katholieke Universiteit Leuven, 2010.

[68] Z. Gordić i C. Ongaro, „Development and Implementation of Orthogonal Planes Images Method for Calibration of Tool Centre Point,“ u *Advances in Robot Design and Intelligent Control*, Cham, 2017.

[69] Z. Gordić i C. Ongaro, „Calibration of robot tool centre point using camera-based system,“ *Serbian Journal of Electrical Engineering*, t. 13, br. 1, pp. 9-20, 2016.

[70] C. Ongaro, „Device and Method for Calibrating of Torch Welding Robot“. Italy Patent TV2014A000165, November 2014.

[71] ABB Robotics, *BullsEye Application Manual, Revision G.*, Vasteras, Sweden, 2012.

[72] J. Hallenberg, „Robot Tool Center Point Calibration using Computer Vision,“ 2007.

[73] F. S. Cheng, „Calibration of Robot Reference Frames for Enhanced Robot Positioning Accuracy,“ u *Robot Manipulators*, M. Ceccarelli, Ur., Rijeka, IntechOpen, 2008.

[74] K. Conrad, P. Shiakolas, S. Edu i T. Yih, „Robotic calibration issues: Accuracy, repeatability and calibration,“ pp. 17-19, August 2000.

[75] Z. Gordić i K. Jovanović, „Identifying Unmodelled Dynamics in Contact Tasks in Industrial,“ u *Proceedings of the 7th International Conference on Electrical, Electronic and*, Etno village “Stanišić“, 2020.

BIOGRAPHY

Zaviša Gordić was born on August 3rd, 1989, in Užice, where he finished Elementary school and Gymnasium. As the best student in generation of the Gymnasium he received Commendations from HRH Prince Aleksandar II Karađorđević. In 2008 he enrolled Bachelor studies at University of Belgrade, School of Electrical Engineering where he graduated in 2016 at the Department of Signals and Systems with an average grade of 9.13, defending the bachelor thesis titled "Application of the four-step commutation method on matrix converter". The same year he started master studies at the same department and finished in 2013 with an average grade of 10.00 and a master thesis titled "Modelling and Control of the Car-Handling Box". In 2013, he enrolled at doctoral studies in robotics on Department for control systems and signal processing, where he passed all exams with average score of 10.00. The primary field of interest and research is related to robot and end-effector calibration, increasing safety in work with industrial robots using collision detection algorithms as well as areas related to human-robot interaction in terms of facilitated programming, control, and use. He is the author of sixteen conference and three journal papers.

Since 2014 he is employed as research apprentice at the University of Belgrade, School of Electrical Engineering on a project of Ministry of Education, Science and Technological Development titled "Research and Development of Ambient Robots with Anthropomorphic Characteristics". In 2015 he was promoted to research assistant on the same project. He finished basic and advanced training for using Denso industrial robots in 2015 in Germany. In 2016 he was engaged on H2020 cascade funding within the scope of ReconCell on a project "Feasibility Study for Serbian Manufacturing Innovation Hub (FS4SMIH)". During 2018 he was involved in ReconCell project titled "Reconfigurable Assembly of Airport Signalization Lights". Since 2019 he is engaged in H2020 project "DIH^2 - A Pan-European Network of Robotics DIHs for Agile Production" as LER (Local Evangelist in Robotics). During 2020 and 2021, as part of the same project, he was enrolled in technological mentoring and evaluation in five technology transfer experiments in Finland, Lithuania, Denmark, Belgium, and Portugal aiming to develop and implement advanced robotics solutions into manufacturing companies. During 2021 he was engaged in H2020 project "BOWI - Boosting Widening Digital Innovation Hubs" as well as on a technological experiment "BrainWatch for increased productivity with improved workers satisfaction" as part of H2020 project "SHOP4CF - Smart Human Oriented Platform for Connected Factories".

In August 2016 he attended the summer school of robotics "IS3 HRC 2016: Italian-Serbian Summer School on Human-Robot Coworking – Master Classes on Human-Robot Coworking and Advanced Robot Grasping", which was held at the Faculty of Mechanical Engineering in Belgrade.

During studies he was honoured with following awards, and scholarships:

- 2017 Best section paper in conference ICETRAN 2017 for the paper titled "Partial Pose Measurements for Identification of Denavit-Hartenberg Parameters of an Industrial Robot"
- 2015 Best section paper in conference ICETRAN 2015 for paper titled "Robot Tool Centre Point Calibration using Analysis of Images from Orthogonal Planes"
- 2013 Awardee of the Scholarship "Dr Zoran Đindjić" of German Eastern Business Association (OA), Federal Ministry for Economic Cooperation and Development (BMZ) and German Corporation for International Cooperation (GIZ)
- 2012. Scholarship holder of IAESTE (International Association for the Exchange of Students for Technical Experience)

образац изјаве о ауторству

Изјава о ауторству

Име и презиме аутора Завиша Гордић

Број индекса 2013/5038

Изјављујем

да је докторска дисертација под насловом

Детекција сила интеракције у индустријској роботици

Detection of Interaction Forces in Industrial Robotics

- резултат сопственог истраживачког рада;
- да дисертација у целини ни у деловима није била предложена за стицање друге дипломе према студијским програмима других високошколских установа;
- да су резултати коректно наведени и
- да нисам кршио/ла ауторска права и користио/ла интелектуалну својину других лица.

Потпис аутора

У Београду, 25. 10. 2022.

Завиша Гордић

Изјава о истоветности штампане и електронске верзије докторског рада

Име и презиме аутора Завиша Гордић

Број индекса 2013/5038

Студијски програм Управљање системима и обрада сигнала

Наслов рада Детекција сила интеракције у индустријској роботици

Ментор Коста Јовановић

Изјављујем да је штампана верзија мог докторског рада истоветна електронској верзији коју сам предао/ла ради похрањивања у **Дигиталном репозиторијуму Универзитета у Београду.**

Дозвољавам да се објаве моји лични подаци везани за добијање академског назива доктора наука, као што су име и презиме, година и место рођења и датум одбране рада.

Ови лични подаци могу се објавити на мрежним страницама дигиталне библиотеке, у електронском каталогу и у публикацијама Универзитета у Београду.

Потпис аутора

У Београду, 25. 10. 2022.

Завиша Гордић

образац изјаве о коришћењу

Изјава о коришћењу

Овлашћујем Универзитетску библиотеку „Светозар Марковић“ да у Дигитални репозиторијум Универзитета у Београду унесе моју докторску дисертацију под насловом:

Детекција сила интеракције у индустријској роботици

Detection of Interaction Forces in Industrial Robotics

која је моје ауторско дело.

Дисертацију са свим прилозима предао/ла сам у електронском формату погодном за трајно архивирање.

Моју докторску дисертацију похрањену у Дигиталном репозиторијуму Универзитета у Београду и доступну у отвореном приступу могу да користе сви који поштују одредбе садржане у одабраном типу лиценце Креативне заједнице (Creative Commons) за коју сам се одлучио/ла.

1. Ауторство (CC BY)

2. Ауторство – некомерцијално (CC BY-NC)

3. Ауторство – некомерцијално – без прерада (CC BY-NC-ND)

4. Ауторство – некомерцијално – делити под истим условима (CC BY-NC-SA)

5. Ауторство – без прерада (CC BY-ND)

6. Ауторство – делити под истим условима (CC BY-SA)

(Молимо да заокружите само једну од шест понуђених лиценци.
Кратак опис лиценци је саставни део ове изјаве).

Потпис аутора

У Београду, 25. 10. 2022.

Завишић Горјана

1. **Ауторство**. Дозвољавате умножавање, дистрибуцију и јавно саопштавање дела, и прераде, ако се наведе име аутора на начин одређен од стране аутора или даваоца лиценце, чак и у комерцијалне сврхе. Ово је најслободнија од свих лиценци.

2. **Ауторство – некомерцијално**. Дозвољавате умножавање, дистрибуцију и јавно саопштавање дела, и прераде, ако се наведе име аутора на начин одређен од стране аутора или даваоца лиценце. Ова лиценца не дозвољава комерцијалну употребу дела.

3. **Ауторство – некомерцијално – без прерада**. Дозвољавате умножавање, дистрибуцију и јавно саопштавање дела, без промена, преобликовања или употребе дела у свом делу, ако се наведе име аутора на начин одређен од стране аутора или даваоца лиценце. Ова лиценца не дозвољава комерцијалну употребу дела. У односу на све остале лиценце, овом лиценцом се ограничава највећи обим права коришћења дела.

4. **Ауторство – некомерцијално – делити под истим условима**. Дозвољавате умножавање, дистрибуцију и јавно саопштавање дела, и прераде, ако се наведе име аутора на начин одређен од стране аутора или даваоца лиценце и ако се прерада дистрибуира под истом или сличном лиценцом. Ова лиценца не дозвољава комерцијалну употребу дела и прерада.

5. **Ауторство – без прерада**. Дозвољавате умножавање, дистрибуцију и јавно саопштавање дела, без промена, преобликовања или употребе дела у свом делу, ако се наведе име аутора на начин одређен од стране аутора или даваоца лиценце. Ова лиценца дозвољава комерцијалну употребу дела.

6. **Ауторство – делити под истим условима**. Дозвољавате умножавање, дистрибуцију и јавно саопштавање дела, и прераде, ако се наведе име аутора на начин одређен од стране аутора или даваоца лиценце и ако се прерада дистрибуира под истом или сличном лиценцом. Ова лиценца дозвољава комерцијалну употребу дела и прерада. Слична је софтверским лиценцама, односно лиценцама отвореног кода.