Robotica: (2020) volume 38, pp. 1717–1736. © Cambridge University Press 2019
doi:10.1017/S0263574719001425

Collision Detection on Industrial Robots in Repetitive Tasks Using Modified Dynamic Time Warping

Zaviša Gordić†* and Kosta Jovanović†

†*School of Electrical Engineering, University of Belgrade, 11000, Belgrade, Serbia*
E-mail: kostaj@etf.rs

(Accepted September 14, 2019. First published online: October 8, 2019)

SUMMARY
This paper presents a non-model-based collision detection algorithm for robots without external sensors and with closed control architecture. A reference signal of repetitive motion is recorded from the robot operation. To detect collisions, the reference is compared with measurements from the robot. One of the key contributions is a novel approach to optimal matching of compared signals, which is ensured by the newly developed modified Dynamic Time Warping (mDTW) method presented in this paper. One of the main improvements of the mDTW is that it 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. In addition to mDTW, two complementary decision rules are developed to identify collisions. The first rule, based on the absolute difference between compared matched samples, uses statistically determined thresholds to perform rapid detection of unambiguous collisions. The second rule is based on Eigen values of the covariance matrix of matched samples, and it employs its higher sensitivity to detect collisions with lower intensity. Results from experimental validation of the proposed collision algorithm on two industrial robots are shown.

KEYWORDS: Collision detection; Industrial robot; Modified dynamic time warping; Human-robot interaction; Elastic similarity measures.

1. Introduction

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.

However, the need for sharing of workspaces of humans and robots becomes more apparent each day. Consequently, lot of attention and effort has been put into finding the best solution for safe and intuitive interaction between them. Modern collaborative and also some lightweight robots with multiple sensors are an exciting field where this interaction opens new possibilities in production, and many interesting research has been performed on them.[1–8] However, it is clear that not all robots in production should or can be collaborative. Furthermore, already installed, conventional industrial robots will be predominant work force for years to come.

Typical industrial robots have much higher performance than their collaborative counterparts, lower price and more robust design. The number of units in existence is on their side as well, since they immensely outnumber collaborative robots. However, they are still required to work in separate areas from humans, as they do not have sensitive collision detection algorithms.

* Corresponding author. E-mail: zavisa@etf.rs

https://doi.org/10.1017/S0263574719001425 Published online by Cambridge University Press

Advances in both conventional industrial robots and collaborative robots are based on sensing and drives, materials technology, mechanical design, as well as controllers and control algorithms. Majority of those advancements lead to mastering safety features. Therefore, it is very hard to distinguish where to draw a line and switch from very efficient conventional industrial robots with additional safety systems to collaborative robots which are still far behind in performance compared with conventional ones.

Prerequisites for safe physical human–robot interaction are described in three nested layers which a robot must guarantee – safety, coexistence and collaboration.[3,7] 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[1,2,9,10] 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 affect 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.[8,11] As such, collaboration is the next from coexistence and is therefore also dependent from reliable collision detection.

A number of successful collision detection algorithms have been proposed so far, and they mostly rely on the model of the robot or some form of dynamics model.[1,3,4,6,7,10,12–15] Some of them include algorithms to identify the model[4,6,7,16] 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. However, such model-based approaches sometimes have difficulties while distinguishing collisions from intentional contacts with external objects, such as ones found in pick and place tasks. In pick and place example, the solution can be to use an external force sensor to account also for the weight and dynamics of the picked object. Alternatively, this information can be incorporated in the form of the model of the object. Various techniques[5,6,13,14,16] have also 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 the algorithms are able to identify collided link[3–5] 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.

One of the basic ideas of the algorithm proposed in this paper 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 a 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 paper and in Ref. [17] 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.

Previous related work[17] 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 when the movement of the robot is modified, in order to determine new limits. Additional tests have also shown the possibility that the algorithm 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

https://doi.org/10.1017/S0263574719001425 Published online by Cambridge University Press

the safety aspect of the system, they may occasionally cause false alarms since the algorithm is not flexible enough to adapt to them.

The main contribution of the paper is the modified Dynamic Time Warping (mDTW) method. Compared to traditional DTW,[18] it offers two advantages of key importance for Collision detection application. Most importantly, unlike,[18–20] it allows matching signals with different contents. That means 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 collision detection concept presented in this paper. At the same time, the mDTW preserves the ability of DTW to optimally match signals by compressing or dilating the time axis. Unlike some other proposed modifications of DTW,[21] the mDTW does not allow skipping samples. This ensures 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.[22] Unlike EDIT distance,[23] Hirschberg's algorithm[24] and Approximate string matching,[23] 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[25] and Discrete Fréchet Distance,[25] the mDTW is intended in 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 allows lowering detection thresholds. This increases the sensitivity of the algorithm and allows detection of lower intensity collisions. The rule based on Eigen values observes distribution of matched sample pairs, offering a new perspective and additional safety measure.

The motivation for this paper 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 paper takes special attention toward 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 toward fulfilling conditions for safety and coexistence.

The second section will present short description of the algorithm and idea behind it. The proposed algorithm's advantages and drawbacks will also be analyzed in context of currently available methods and achievements.

Third section provides a brief analysis of the character of the monitored signal, along with its dynamics and features. It will explain which features of the signal can be used as reliable indicators of detected collisions, as well as some of the most common reasons for false detection.

The fourth section explains DTW method, as one of the most efficient ways for matching two time sequences.[18, 19] Additionally, it presents the modification of DTW which was developed for the purpose of the collision detection algorithm itself. Modification brings the benefit of being able to match subsequence to the corresponding section of the complete sequence, making the modified DTW applicable in online collision detection, rather than only in offline analysis.

Considerations about the collision detection rules are presented in the fifth section in conjunction with conclusions from statistical analysis of the signal. One detection rule is based on the changes in Eigen values of the observed set of measurements optimally matched with corresponding part of the reference sequence. Complementary detection rule is based on observing the maximum difference between matched pairs.

The sixth section presents experimental validation of collision detection algorithm along with the performance analysis. Experiments are conducted on Denso VP-6242 and ABB IRB120 industrial robots in order to verify not only the performance of the algorithm itself but also its possibility for implementation on different robot brands.

Conclusions from the process of design and implementation, as well as experimental validation, are presented in the seventh section. The section also provides overview of benefits that the presented algorithm brings, together with some of its drawbacks and directions for future work.

https://doi.org/10.1017/S0263574719001425 Published online by Cambridge University Press

2. Principal Design of the Collision Detection Algorithm

This paper is dedicated to improving the performance and flexibility of previously developed algorithm.[17] Due to increased complexity of the algorithm presented in this paper, the possibility of its full integration on robot's controller is reduced for many brands because of their closed architecture and lack of interface.[14] 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 implementability, it was decided to use Ethernet TCP/IP connection, since it is a standardized and widely available communication connection type.

Overall structure and behavior of the collision detection system are designed to be simple in order to minimize reaction time and load on the robot side.

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 algorithm records a certain number of measurements and uses them as reference signal. Otherwise, if the algorithm is actively detecting collisions, then 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 corresponding section of the reference. Using matched signals, collision detection rules decide whether or not the collision occurred on any of the robot joints.

A signal indicating whether or not 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 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. 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 behavior over time. It is important that the reference signal contains more than one cycle of recorded repetitive robot movement. This ensures that transition from one cycle of operation 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 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 before, reference signal 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. Sampling frequencies of reference signal and measurement vector do not necessarily need to be identical, as it will be explained in the third section in more detail together with some other issues. To overcome this situation, a modification of DTW Algorithm (mDTW) was developed and presented in the fourth section. It allows online matching of reference and measurement vector and enables application of collision detection rules.

Collisions of robot are identified based on two detection rules, described in the fifth section. 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

https://doi.org/10.1017/S0263574719001425 Published online by Cambridge University Press

analysis of the measurements without collisions. As described in the fifth section, thresholds for decision rules are set as values 20% higher than absolute differences occurring during matching of measurement vector without collisions.

With regard 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.[7–9,12–14] Although fully recognizing the benefit of refined collision reaction strategies, this paper focuses more on reliable detection and its implementability across various brands. As such, the algorithm proposed in this paper will employ stopping upon detected impact as the simplest reaction strategy which is also available on vast majority of robot brands and generations.

Besides all the benefits the proposed collision detection algorithm possesses, there are some drawbacks which need to be considered. First is related to the fact that, unlike model-based algorithms,[1,3,4,6,7,10,12–15] it can only work for repetitive tasks. Every modification of the trajectory or speed at which the repetitive motion is performed should be followed by recording a new reference signal and setting thresholds for decision rules. The same is also valid in situations when, for example, the object of manipulation is changed in manipulation tasks, or if different assembly is performed. Although such modifications can be made with small amount of time and effort, they do require some human operator understanding and involvement. Furthermore, unlike similar algorithm presented in Ref. [17], this collision detection algorithm requires permanent connection with a PC or some similar processing unit. The reason for this is increased complexity, which means it cannot be fully implemented in 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.

Although the developed mDTW presented in this paper 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 signal 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 axes 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.

3. Signal Analysis and Considerations

The main idea during design phase of the collision detection algorithm was to make it as versatile and implementable as possible, on variety of robot brands and generations. Consequently, signals chosen to work with also needed to be readily available. Most commonly, chosen signals are either current measurements or torque estimations based on current measurements.[14] Some newer robots possess torque sensors, and consequently they can provide torque measurements directly.[1–8] However, this is not a common situation in majority of conventional robots used in industry.

On the other hand, current-based torque estimations can be obtained in majority of robots directly, and the presented algorithm is analyzed 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 estimation in the rest of the paper. 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.

It is important to emphasize that the algorithm presented in this paper 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

https://doi.org/10.1017/S0263574719001425 Published online by Cambridge University Press

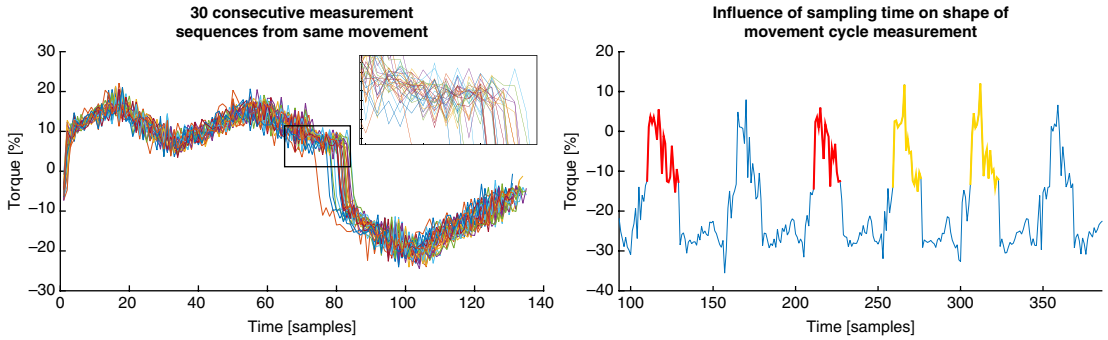


Fig. 1. (left) Recorded torque sequences from 30 consecutive executions of the same movement task. Enlarged section shows that sequences are shifted in time and also that samples differ in value in corresponding time instants. (right) Six consecutive movement cycles and differences between them which are caused by sampling. Signal peaks shown in sections of signal marked in yellow are not visible on sections marked in red.

easier to comprehend. Torques have a more physical manifestation, making them more understandable. Additionally, current measurements provided by Denso are in absolute values, which means that graphs would not clearly illustrate the nature of the movement.

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.

As a consequence, it is very difficult to guarantee that measurements obtained in one of the 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. Additionally, it is clear that one cycle of robot movement does not 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 Fig. 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.

On model-based algorithms, the sampling time is not a major issue, since both commanded current/torque and measurement can be acquired at the same time, making their comparison easier. However, in proposed algorithm, the reference signal is not generated by the model, but is instead a recorded sequence. Therefore, previously mentioned realization of parallel tasks imposes a new set of related challenges and issues related to comparison of corresponding signal values. The challenges and issues are tackled in the following section.

Other important aspects to consider are short duration peaks which may occur within one cycle of movement execution as a result of normal behavior of the robot, such as the ones shown on the right graph of Fig. 1. Because of uneven sampling, these peaks can sometimes be sampled at different times, making their intensity different, or the peaks themselves may even be completely missed, potentially leading to false collision detection. Collision detection algorithm which utilizes such measurements needs to be invariant to their time sampling and compensate for their time compression or dilatation.

4. Modified Dynamic Time Warping

One of the main contributions of this paper is a modification of the 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. To the authors' best knowledge, 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[23]

https://doi.org/10.1017/S0263574719001425 Published online by Cambridge University Press

distance were one of the first to be analyzed, since it is 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[24] and Approximate string matching.[23] 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[25] and Discrete Fréchet Distance[25] were also analyzed due to their versatility. Nevertheless, the key drawback of such methods was that they were intended for parametric curve comparison, rather than time series.

DTW[18] 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.[19] 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):

$$d(1,\,1) = |a(1) - b(1)|\,,\; i = 1, j = 1 \tag{1}$$

$$d(1, j) = |a(1) - b(j)| + d(1, j-1),\; i = 1,\; 1 < j < n \tag{2}$$

$$d(i,\,1) = |a(i) - b(1)| + d(i-1,\,1),\; 1 < i \le m, j = 1 \tag{3}$$

$$d(i, j) = |a(i) - b(j)| + \min\left(d(i-1, j-1), d(i, j-1), d(i-1, j)\right)\,,\; 1 < i \le m,\; 1 < j \le n \tag{4}$$

In the aforementioned expressions, m and n stand for lengths of the first and second signal, respectively. 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. Analog 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 $d(m,n)$.

$$\min\left(d(i-1, j-1), d(i, j-1), d(i-1, j)\right)\,,\; i \neq 1, j \neq 1 \tag{5}$$

$$d(i, j-1),\; i = 1, j \neq 1 \tag{6}$$

$$d(i-1, j),\; i \neq 1, j = 1 \tag{7}$$

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 neighboring 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 toward the first element, belonging to the 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

https://doi.org/10.1017/S0263574719001425 Published online by Cambridge University Press

two aforementioned observations lead to the conclusion that the contents, or shapes, of the compared signals need 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, that is, 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[19] 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,[22] but this type of research is not a subject of this paper.

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, or allow skipping some samples, such as in Minimal Variance Matching.[21] 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 reinitiated 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.

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 taken into account.

A modification of DTW is proposed in this paper which 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).

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

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

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

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

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 neighboring 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, $\min(d(m,\mathrm{j}))$, $1 = \mathrm{j} = n$. The search continues with finding the minimum of preceding values as described with (12) and (13) and stops immediately when the first row of the matrix is reached.

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

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

https://doi.org/10.1017/S0263574719001425 Published online by Cambridge University Press

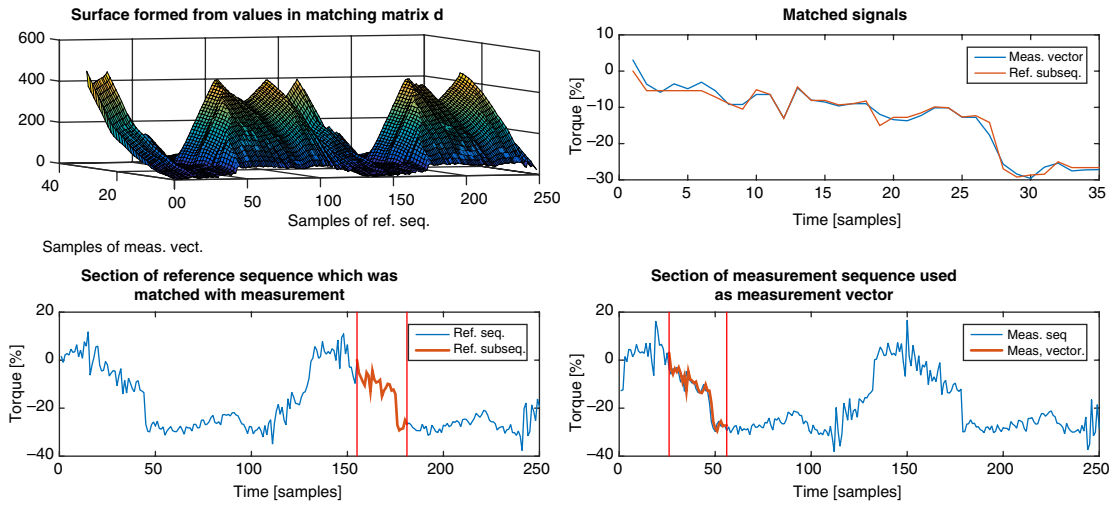


Fig. 2. Signals matching performed by modified DTW (mDTW). (upper left) Values of matrix used for optimal matching. Optimally matched pairs of signals correspond to the lowest sections of the surface. (upper right) Two matched signals – blue signal corresponds to the section of the reference signal, 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.

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 paper, a reference signal to which all other signals are compared is called the reference sequence. The reference sequence is compared in this paper 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 signal 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 shown on Fig. 2 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 Fig. 3.

Similar to the solution described in Refs. 19, 20, the modification proposed in this paper limits the signal lengths, and consequently the computational complexity, by using constant lengths of the reference signal 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 MVM,[21] mDTW method does

https://doi.org/10.1017/S0263574719001425 Published online by Cambridge University Press

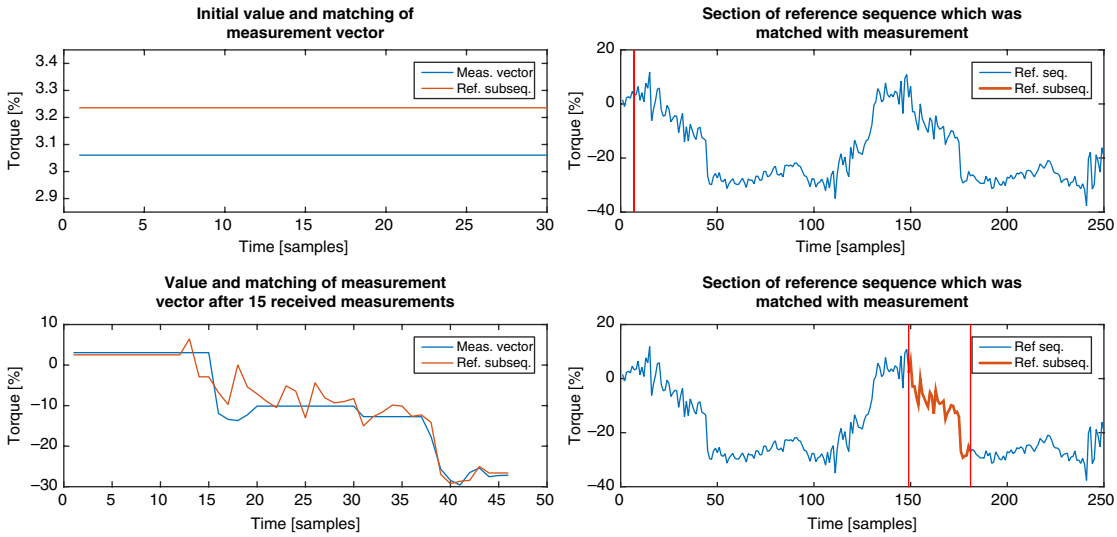


Fig. 3. Two instants during formation of the 30 samples long measurement vector. (upper left) Initially, all values in the measurement vector are the same and equal to the first measured value received from the robot, Therefore, they are matched with single value of the reference sequence. (upper right) Point of the reference sequence paired with the measurement vector. (lower left) Measurement vector shown in blue color 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.

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[20] 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 are the same.

5. Decision Rules 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 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 values 20% higher than maximum absolute difference of samples which did not belong to a collision, similarly as implemented in.[14] 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 Fig. 4.

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 Fig. 5. 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

https://doi.org/10.1017/S0263574719001425 Published online by Cambridge University Press

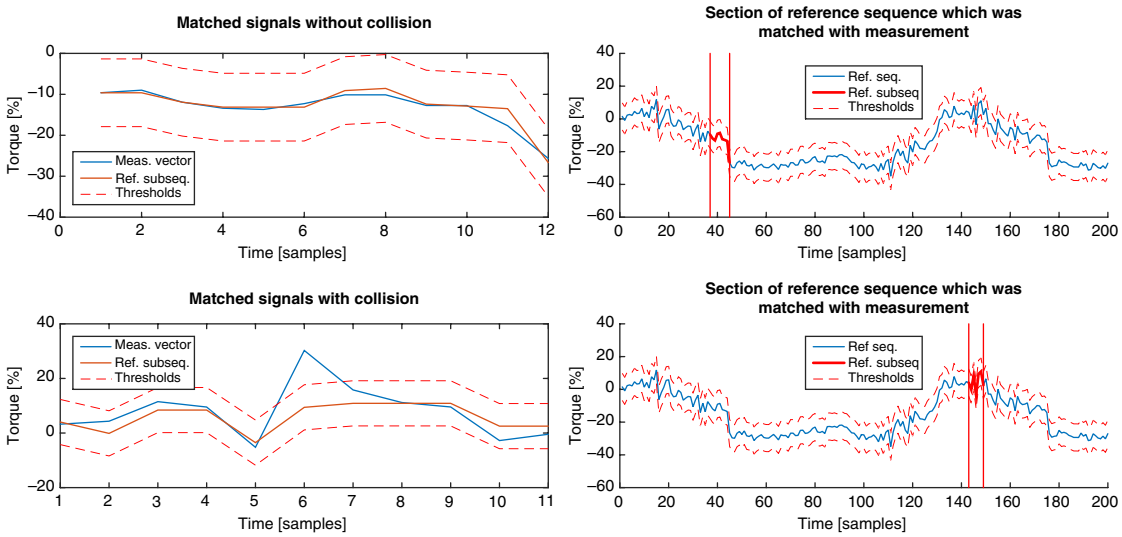


Fig. 4. Matched signals on second axis of the robot. (upper left) Section of the reference signal matched with the measurement vector without collisions as well as signal thresholds imposed by the first rule. (upper right) Subsequence of the reference signal to which the measurement from the upper left 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.

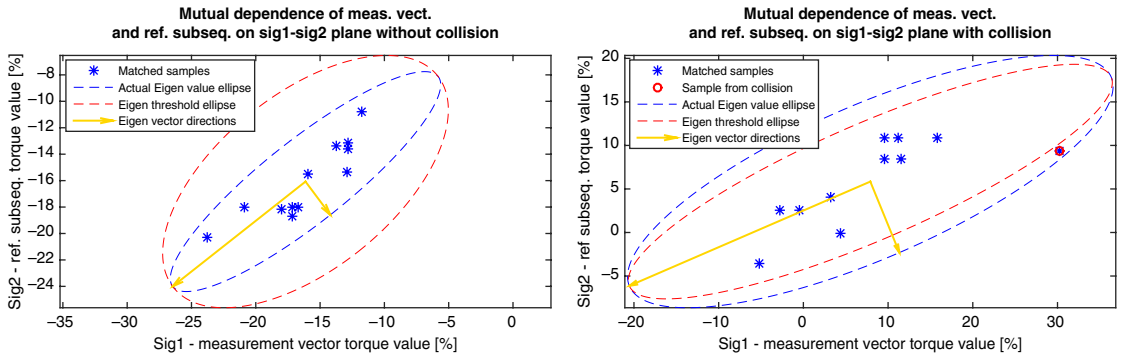


Fig. 5. (left) Approximately linear distribution of paired samples without collision on sig1–sig2 plane. Higher intensity Eigen vector approximately corresponds to the line fitted to the measurements. The lower intensity Eigen vector is a measure of dissipation of sample pairs along direction orthogonal 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.

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. Similar to the first 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.

https://doi.org/10.1017/S0263574719001425 Published online by Cambridge University Press

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 behavior to collision. While each rule independently has its own benefits, the highest performance in collision detection is reached when they are used simultaneously.

6. 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 goal of online testing was to experimentally validate the performance of the collision detection algorithm.

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 of 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 that 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 summarizes results from both offline and online testing. It also comments the benefits and drawbacks of the algorithm concept as well as of decision rules.

6.1. Offline testing

The collision detection algorithm presented in this paper 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 one 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

https://doi.org/10.1017/S0263574719001425 Published online by Cambridge University Press

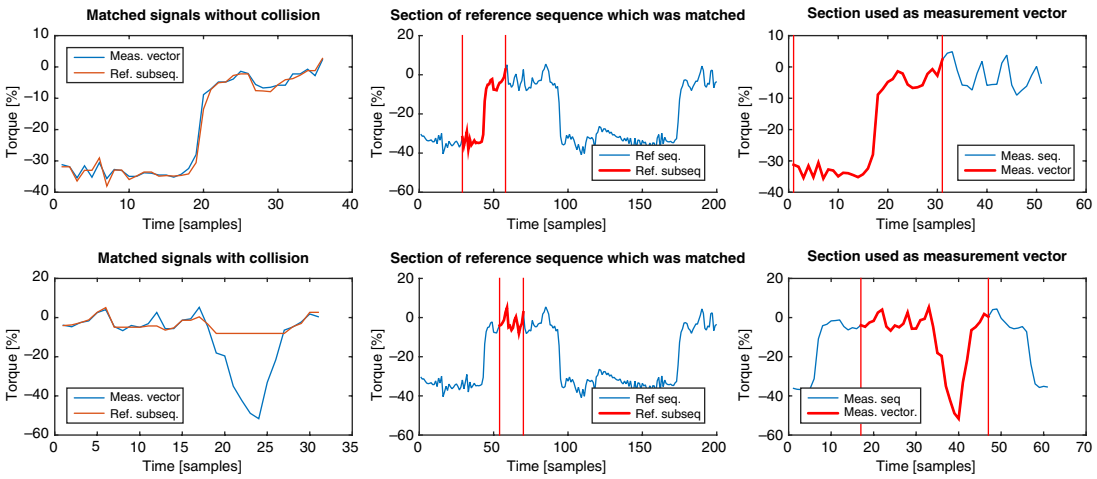


Fig. 6. Testing of modified DTW, example on the third axis. (upper left) Measurement vector matched to the subsequence of the reference sequence. (upper middle) Matched subsequence of the reference sequence. (upper right) Portion of measurement sequence from which the measurement vector was formed. (lower 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.

matching regardless of whether the measurement sequence was taken from the set of measurements recorded with or without collision. Figure 6 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 times. The measurement sequence was resampled with up to three 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 Fig. 7. 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.

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 I. 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 20% higher than all differences characterized as normal behavior of the system for that particular axis. 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 Fig. 8, while the statistical analysis of the first detection rule can be seen in Table I.

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 that 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 performance of the second rule are presented in Table I. 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 to 0.65 s to identify the collision). Figure 9 demonstrates

https://doi.org/10.1017/S0263574719001425 Published online by Cambridge University Press

Table I. Statistical results from the offline testing.

	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%

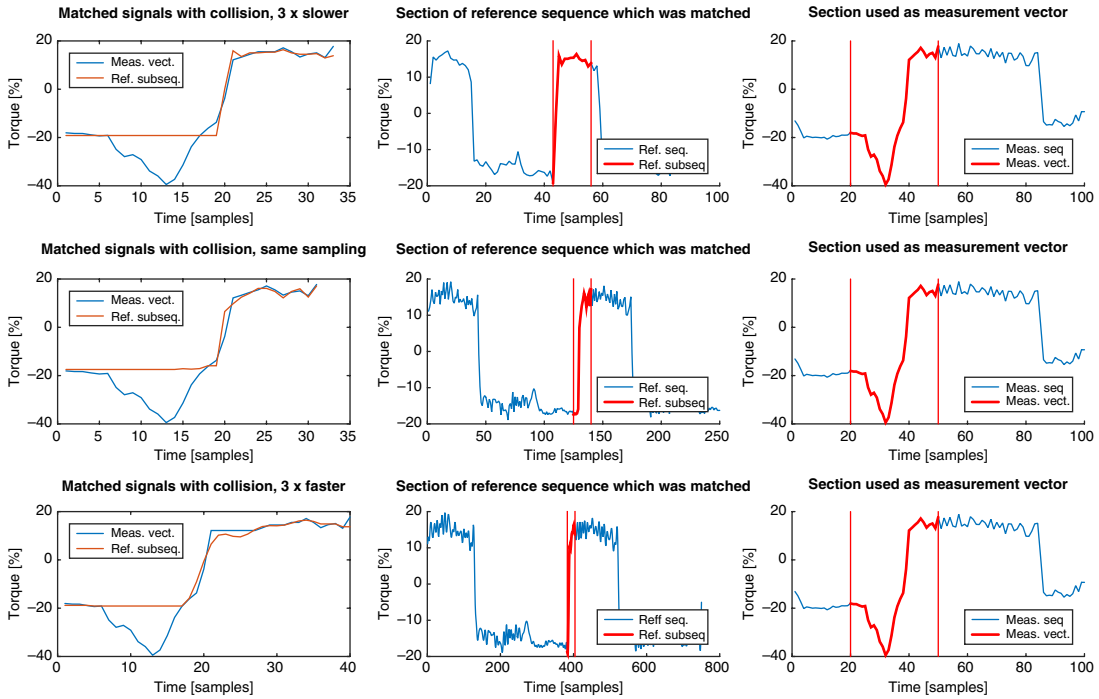


Fig. 7. Influence of sampling frequency on the quality of matching on the fourth axis. (upper row) Reference sequence sampled at three times lower frequency compared to measurement sequence. (middle row) Reference sequence sampled at the same frequency as measurement sequence. (lower row) Reference sequence sampled at three times higher frequency compared to measurement sequence. (left column) Matched samples of measurement vector and reference subsequence with different sampling times. (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.

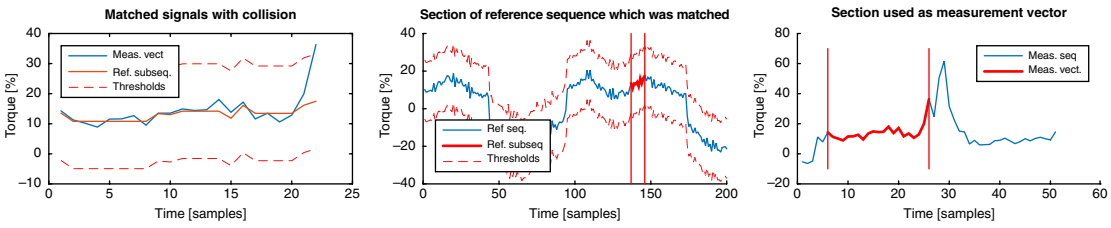


Fig. 8. Example of the first rule application on the first robot axis. (left) Sample of measurement vector has crossed the threshold, triggering the collision detection. (middle) Subsequence of the reference to which the measurement vector was matched. (right) Portion of the measurement sequence used to form the measurement vector.

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 10 shows that the collision would be detected also by the first rule, but one cycle after the second rule was activated. With regard to reaction times, this is very significant, and although such situations are not common, they are very important.

https://doi.org/10.1017/S0263574719001425 Published online by Cambridge University Press

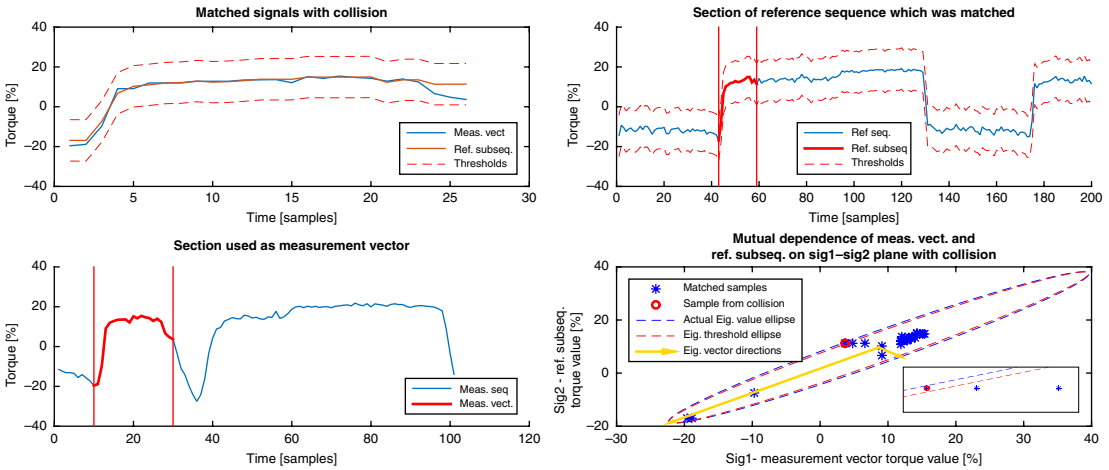


Fig. 9. Example of the second rule application on fourth robot axis in moment when the first rule was not yet triggered. (upper left) Sample of measurement vector has not yet crossed the threshold of first rule. (upper right) Subsequence of the reference to which the measurement vector was matched. (lower 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.

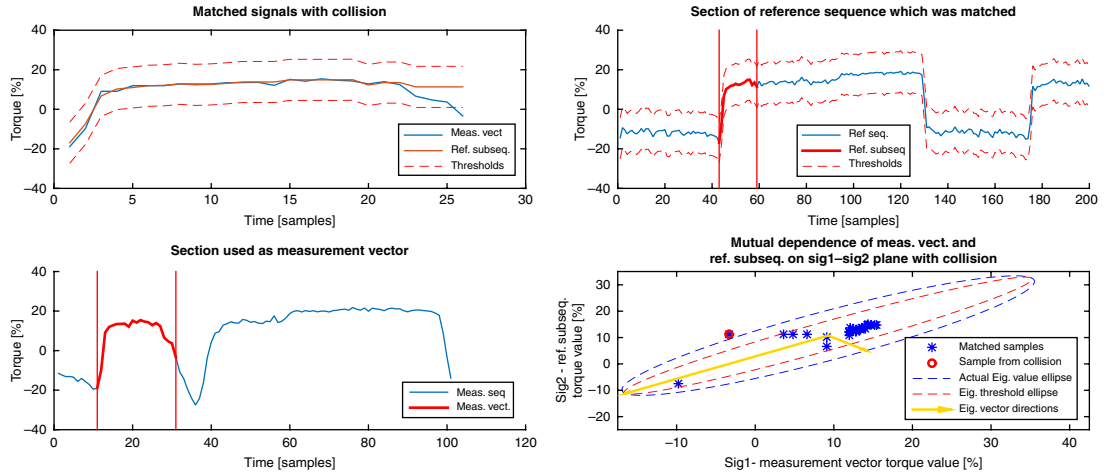


Fig. 10. Example of the first rule being triggered for collision on the fourth robot axis one sample after collision was detected by second rule. (upper left) Sample of measurement vector has crossed the threshold of the first rule after one sample more compared to the moment shown on Fig. 9. (upper 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.

6.2. Online testing

Online testing was conducted with two 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 Fig. 11. Videos of online testing are available on https://youtu.be/gEysqZhN7Y0 for Denso and on https://youtu.be/_SXtQOEJM_I for ABB robot.

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

https://doi.org/10.1017/S0263574719001425 Published online by Cambridge University Press

Table II. Statistical results from the online testing.

	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%



Fig. 11. Robots used in experimental validation. (left) Denso VP-6242 robot used for offline and online testing. (right) ABB IRB120 robot which was used in online testing.

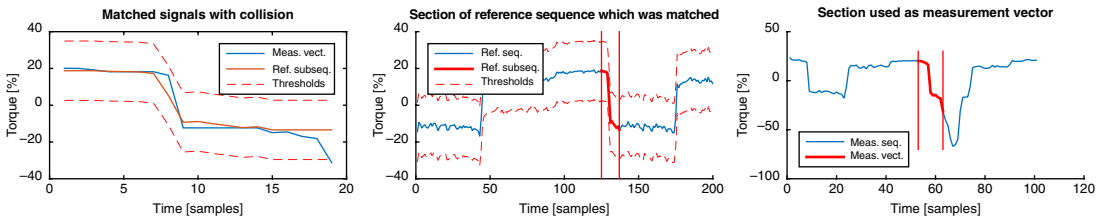


Fig. 12. Results from online collision detection on the fourth axis of Denso robot. (left) collision detected using application of the first rule. (middle) Subsection of the reference sequence to which the measurement vector was matched. (right) Section of measurement sequence used as measurement vector.

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 II. Figure 12 shows one example of a collision detected on the fourth axis of Denso robot using the first rule. Collision detection example on the first axis of Denso robot, in which only the second detection rule was activated one cycle before the first rule detected it, is shown on Fig. 13.

Experimental results from online testing on ABB robot are shown in figures below. Figure 14 shows detection of a collision on the first axis of ABB robot, while Fig. 15 shows detection of the third axis of the same robot.

https://doi.org/10.1017/S0263574719001425 Published online by Cambridge University Press

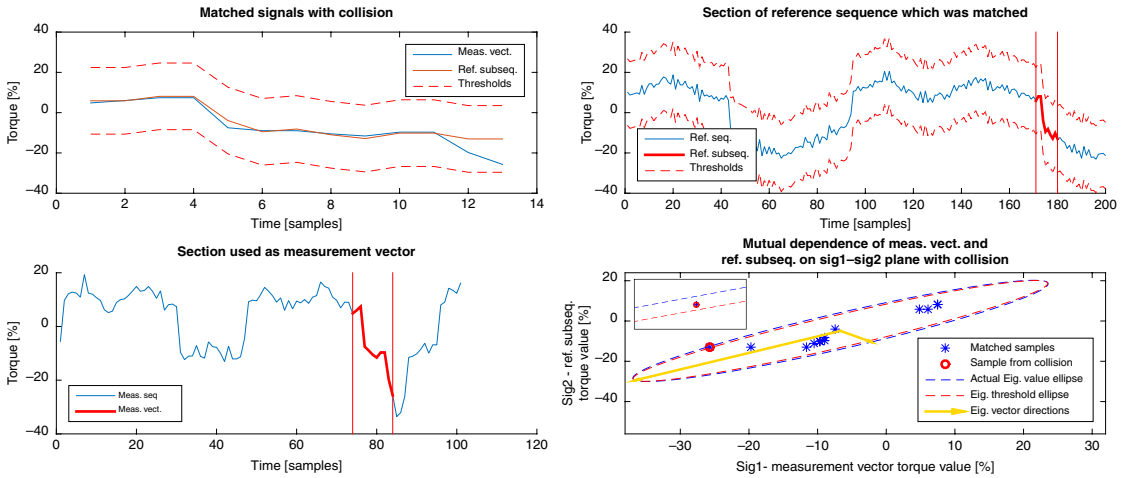


Fig. 13. Results from online collision detection on Denso VP-6242 robot, in which the second rule detected the collision on first axis one sample before the first rule detected it. (upper left) Collision is not yet detected using first rule. (upper right) Subsequence of the reference sequence to which 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.

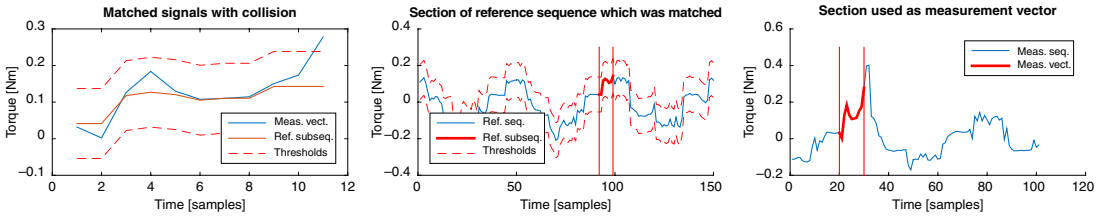


Fig. 14. Results from online collision detection on ABB IRB120 robot, example from the first axis. (left) Collision detected using application of the first rule and matched signals. (middle) Subsequence of the reference sequence to which the measurement vector was matched. (right) Section of measurement sequence used as a measurement vector.

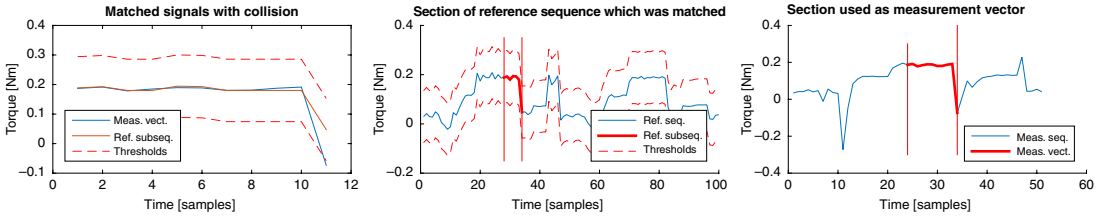


Fig. 15. Results from online collision detection on ABB IRB120 robot, example on the third axis. (left) Collision detected using application of the first rule on the third axis of the robot. (middle) Subsequence of the reference sequence to which the measurement vector was matched. (right) Section of 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 includes also the communication time between the robot and the PC.

https://doi.org/10.1017/S0263574719001425 Published online by Cambridge University Press

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

6.3. 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 Tables I and II, 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 behavior, the two implemented decision rules complement each other. Testing on a large sample has shown that zero collisions were classified as nominal behavior 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 Tables I and II 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 only on a statistical analysis of the signal and heuristics but also allow for some smart adaptability. There are also possibilities for designing new rules, or extending and improving existing ones. In all cases, complexity of the algorithm should be carefully analyzed 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 or not 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.

7. Conclusion

The intention of this paper was to present a collision detection algorithm as a contribution toward safer workspace sharing and future collaboration between humans and industrial robots without external sensors or open control architecture. Compared to most existing methods, the collision detection algorithm presented in this paper is not based on a robot model. Instead, it relays 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.

As a corner stone for reliable collision detection, this paper proposed an mDTW which was used in order to overcome sampling uncertainties imposed by the robot controller design. As an important improvement of DTW, the proposed modification enables comparing measurements with the

https://doi.org/10.1017/S0263574719001425 Published online by Cambridge University Press

most similar section of the reference signal, rather than having to compare them with the entire signal. Partial matching enables online application of time warping principles and limits time and computation resources needed to perform matching.

On the basis of optimal matching of signals, two complementary decision rules were designed to detect a potential collision. The first rule, based on the absolute difference between compared matched samples, uses statistically determined thresholds to perform rapid detection of unambiguous collisions. The second rule that is based on Eigen values of the covariance matrix of matched samples employs its higher sensitivity to detect collisions with lower intensity.

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 paper, 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 intention of the paper, 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, future work on the algorithm itself will include reducing 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 recenttly presented in,[20] and its adaptation to the specifics of the mDTW has a potential for great improvements in computational requirements and reaction speed. Other related work will include improving the flexibility of the algorithm and its capability to adapt to a family of similar movements without having to reset reference values. To this end, solutions such as ones presented in Ref. 20 offer new insights on how the elasticity of the similarity measure can be dealt with when the sequence under comparison evolves over time.

Acknowledgment

The work on this project was partly supported by the Ministry of education, science, and technological development, Republic of Serbia, grant No. TR35003, and by DIH^2 project from the European Union's Horizon 2020 research and innovation program under grant agreement No. 824964.

Supplementary Material

To view supplementary material for this article, please visit https://doi.org/10.1017/S0263574719001425

References

1. A. De Luca, A. Albu-Schäffer, S. Haddadin and G. Hirzinger, "Collision Detection And Safe Reaction With The DLR-III Lightweight Manipulator Arm," *IEEE/RSJ International Conference on Intelligent Robots and Systems – IROS*, Beijing, China (2006) pp. 1623–1630.
2. S. Haddadin, A. Albu-Schäffer, A. D. Luca and G. Hirzinger, "Evaluation of Collision Detection and Reaction for a Human-Friendly Robot on Biological Tissues," *IARP International Workshop on Technical Challenges and for Dependable Robots in Human Environments*, Los Angeles, CA, USA (2008).
3. A. De Luca and F. Flacco, "Integrated Control for pHRI: Collision Avoidance, Detection, Reaction and Collaboration," *IEEE RAS & EMBS International Conference on Biomedical Robotics and Biomechatronics – BioRob*, Roma, Italy (2012) pp. 288–295.
4. F. Dimeas, L. Avendaño-Valencia and N. Aspragathos, "Human – robot collision detection and identification based on fuzzy and time series modelling," *Robotica* **33**(9), 1886–1898 (2015).
5. A.-N. Sharkawy, P. Koustoumpardis and N. Aspragathos, "Manipulator Collision Detection and Collided Link Identification based on Neural Networks," *International Conference on Robotics in Alpe-Adria Danube Region – RAAD*, Patras, Greece (2018).
6. P. Pastor, L. Righetti, M. Kalakrishnan and S. Schaal, "Online Movement Adaptation Based on Previous Sensor Experiences," *IEEE/RSJ International Conference on Intelligent Robots and Systems – IROS*, San Francisco, CA, USA (2011) pp. 365–371.

https://doi.org/10.1017/S0263574719001425 Published online by Cambridge University Press

7. F. Flacco and A. De Luca, "Safe Physical Human-Robot Collaboration," *IEEE/RSJ International Conference on Intelligent Robots and Systems – IROS*, Tokyo, Japan (2013) p. 2072.
8. E. Magrini, F. Flacco and A. De Luca, "Control of Generalized Contact Motion and Force in Physical Human-Robot Interaction," *IEEE International Conference on Robotics and Automation – ICRA*, Seattle, Washington, USA (2015) pp. 2298–2304.
9. S. Haddadin, A. Albu-Schäffer and G. Hirzinger, "Safe Physical Human-Robot Interaction: Measurements, Analysis and New Insights," **In:** *Robotics Research, Springer Tracts in Advanced Robotics* (M. Kaneko and Y. Nakamura, eds.), vol. 66 (Springer, Berlin, Heidelberg, 2010) pp. 395–407.
10. S. Haddadin, A. Albu-Schäffer and G. Hirzinger, "Safety evaluation of physical human-robot interaction via crash-testing," **In:** *Robotics: Science and Systems* (MIT Press, Cambridge, Massachusetts, 2007) pp. 217– 224.
11. J. Heinzmann and A. Zelinsky, "Quantitative safety guarantees for physical human-robot interaction," *Int. J. Robot. Res.* **22**(7–8), 479–504 (2003).
12. S. Haddadin, A. Albu-Schäffer, A. De Luca and G. Hirzinger, "Collision Detection and Reaction: A Contribution to Safe Physical Human Robot Interaction," *IEEE/RSJ International Conference on Intelligent Robots and Systems – IROS*, Nice, France (2008) pp. 3356–3363.
13. A. De Luca and R. Mattone, "Sensorless Robot Collision Detection and Hybrid Force/Motion Control," *IEEE International Conference on Robotics and Automation – ICRA*, Edmonton, Alberta, Canada (2005) pp. 1011–1016.
14. M. Geravand, F. Flacco and A. De Luca, "Human-Robot Physical Interaction and Collaboration Using an Industrial Robot With a Closed Control Architecture," *IEEE International Conference on Robotics and Automation – ICRA*, Karlsruhe, Germany (2013) pp. 4000–4007.
15. Y. Yamada, Y. Hirasawa, S. Huand and Y. Umetani, "Fail-Safe Human/Robot Contact in the Safety Space," *IEEE International Workshop on Robot and Human Communication – RO-MAN*, Tsukuba, Ibaraki, Japan (1996) pp. 59–64.
16. M. Erden and T. Tomiyama, "Human-intent detection and physically interactive control of a robot without force sensors," *IEEE Trans. Robot.* **26**(2), 370–382 (2010).
17. Z. Gordić and K. Jovanović, "Fully Integrated Torque-Based Collision Detection in Periodic Tasks for Industrial Robots with Closed Control Architecture," *International Conference on Robotics in Alpe-Adria Danube Region – RAAD*, Patras, Greece (2018).
18. M. Müller, "Chapter 4: Dynamic Time Warping," **In:** *Information Retrieval for Music and Motion* (Springer-Verlag, Berlin, Heidelberg, 2007) pp. 69–84.
19. I. Oregi, A. Pérez, J. Del Ser and J. A. Lozano, "On-Line Dynamic Time Warping for Streaming Time Series," *Joint European Conference on Machine Learning and Knowledge Discovery in Databases – ECML/PKDD*, Skopje, Macedonia (2017) pp. 591–605.
20. I. Oregi, A. Pérez, J. Del Ser and J. A. Lozano, "On-line elastic similarity measures for time series," *Pattern Recog.* **88**, 506–517 (2019).
21. L. J. Latecki, V. Megalooikonomou, Q. Wang, R. Lakaemper, C. A. Ratanamahatana and E. Keogh, "Partial Elastic Matching of Time Series," *IEEE International Conference on Data Mining – ICDM*, Houston, Texas, USA (2005) pp. 577–584.
22. D. Sart, A. Mueen, W. A. Najjar, E. J. Keogh and V. Niennattrakul, "Accelerating Dynamic Time Warping Subsequence Search with GPUs and FPGAs," *IEEE International Conference on Data Mining – ICDM*, Sydney, Australia (2010) pp. 1001–1006.
23. G. Navarro, "A guided tour to approximate string matching," *ACM Comput. Surv.* **33**(1), 31–88 (2001).
24. D. S. Hirschberg, "A linear space algorithm for computing maximal common subsequences," *Commun. ACM* **18**(6), 341–343 (1975).
25. C. F. Jekel, G. Venter, M. P. Venter, N. Stander and R. T. Haftka, "Similarity measures for Identifying material parameters from hysteresis loops using inverse analysis," *Int. J. Mater. Form.* **12**(3), 355–378 (2019).

https://doi.org/10.1017/S0263574719001425 Published online by Cambridge University Press