# Trajectory-Based Exception Learning Strategy in the Context of Robotics

**Ron Goerz, Mihael Simonič, Aleš Ude, Matevz Majcen Horvat**

*Institute Jožef Stefan E1, Jamova cesta 39, SI-1000 Ljubljana*
*E-Mail: a01527267@unet.univie.ac.at*

## Trajectory-Based Exception Learning Strategy in the Context of Robotics

***Abstract.*** *Human intervention is often needed if an assembly task which is performed by a robot arm fails. Through human-guided kinesthetic teaching, such error cases can be solved. To solve them sustainably, the error cases need to be classified, such that similar situations can be handled autonomously by the robot. One approach to tackle this problem is to use recorded trajectories. This article first compares two trajectory-averaging methods, based on arithmetic mean and Dynamical Time Warping Barycenter Averaging (DBA). In the second step a novel classification method, based on Gaussian Radial Basis Functions (RBF), is compared with a well-established Dynamical Time Warping (DTW) distances-based method, with the purpose of assigning new thought trajectories correctly. The methods are applied to a database representative of errors occurring in a peg-in-hole task. It turns out that DBA averages a set of trajectories qualitatively better than the arithmetic mean and can tackle the multiple-modes problem. DBA in combination with a DTW distance-based Similarity Measure (SM) provides the best results, whereas the novel RBF-based method is not reliable.*

## 1 Introduction

This article focuses on providing suitable methods to enhance the autonomy of industrial robots. It builds on the frameworks of Nemec et al. [1] and Simonič et al. [2], who develop an exception strategy for assembly tasks. If an error occurs during such an assembly task e.g., through small changes in the robot's environment, human intervention is required. However, most of the current systems are not able to learn from the previous situations. Therefore human intervention is needed again, even if the same or a similar situation repeats [1].

Nemec et al. [1] developed an exception strategy learning, through which a robot is increasingly able to resolve errors autonomously. If the robotic system recognizes an error, it switches into the gravity compensation mode. In this mode, an operator can move the arm of the robot manually by using kinesthetic teaching [2]. After an internal database is obtained with corrective actions and the errors are contextualized, a suitable action is computed using statistical methods [3]. The error context is determined from visual and force-torque sensor data, which are combined in the process of data fusion. The resulting high-dimensional context description is further reduced to a low-dimensional one using predictive clustering trees [3]. Based on the context estimation, the robot system can categorize the new error into one of the previously seen *error types*. This model works well as long as the new error lies in the domain in which the classifier was trained previously. As not all the sensor information is captured in the low-dimensional representation, it can happen, that the context estimation will be the same for two different error types. The problem arises, that a novel occurring type of error is classified as an existing error type. Another problem in this context is, that human demonstrations can differ significantly from each other for one error type, which is known as the *multiple-modes problem* and is described in more detail in [4].

This article compares and provides suitable methods in two steps. To tackle the above problems the variability of human-demonstrated corrective actions is used. As a first step, it is necessary to find a method, which leads to a suitable representation of one error type, to then be able to distinguish them in the second step. For this purpose, the positional components in Cartesian space and the orientational components in unit quaternions at each time point of a set of corrective actions can be used to determine an error type. One of these error types can be represented by the average of the trajectories of the corrective actions, in the following called *correction trajectories* and *average trajectory*.

To find a suitable averaging method for a set of correction trajectories, two different approaches are compared in the first step. The first approach uses the *arithmetic mean* to average a number of corrective trajectories, regarding their positional component and a quaternion conform method for the orientational component. As a second approach, *Dynamical Time Warping Barycenter Averaging* is used to average the corrective trajectories. This type of averaging is time-independent and highly robust, which leads to different resulting average trajectories [5].

The second step demands a method, capable of deciding to which error type a new corrective action belongs. This can be done by comparing the trajectories of the new corrective action with the average trajectories of the existing error types. For that purpose, a suitable *similarity measure* is needed which provides the basis for the classification. Related works use such similarity measures based on, e.g. Euclidian distances [6]. In this article, two different methods are compared. The first method uses the distances of *Dynamical Time Warping* as a similarity measure and is well-established in the context of comparing time series [5], [7]. In a novel second approach, the weights of *Gaussian Radial Basis Functions* are used for this purpose. In [8] it is mentioned,

that trajectories, which are encoded in a Gaussian Mixture Model, which uses Gaussian kernel functions can be used for classification. The method is well known for interpolation and seemed to be a promising approach in the beginning. It has the advantages, that it is time-independent, the orientational component is considered in the form of unit quaternions and the amount of kernel functions can be set to the same number for different trajectories, through which the weights get comparable [9]. This comparison can be used to determine a similarity between two trajectories in Cartesian space.
To evaluate and compare the different methods, a database with 12 different representative error types was created [10]. The different methods of the first and second steps as well as the structure of the database are described in the following sections.

## 2 Methods

In this section, first, the structure and the labeling of the database are explained. This is followed by two separate steps. The first step describes the two averaging methods based on the arithmetic mean and DBA. In the second step, the two methods for the similarity measures are presented. One is based on DTW distances, while the other is based on the weights of Gaussian RBFs.

### 2.1 Database

To build up a representative database [10], capable of evaluating the different methods, existing infrastructure at the Institute Jožef Stefan was utilized. This includes the whole set-up of a Franka Emika Research 3 robot arm with an ATI Nano 25 force/torque sensor and an Intel Realsense D435 RGB-D camera. To take control of the robot arm, the intern software, developed by Žlajpah et al. [11] is used. The database contains data, obtained from executions of a peg-in-hole task, similar to the Cranfield Benchmark [3]. In the setup, the robot should put a peg in a hole, whereby systematic errors with an offset of 10mm in the *x*- and *y*-direction are made (see Figure 1). During the task, one error trajectory and ten correction trajectories are recorded for each of the 12 different error types. With regard to the correct insertion position of the peg in the hole, a class is defined as the positional (*x* and *y* direction) and orientational (around *x*, *y* and *z* axes) offset of a specified error position (see Figure 1). Each entry includes a trajectory, force-torque sensor data, an RBG image sequence (sampling rate 10Hz) and a point-cloud sequence (sampling rate 10Hz). A trajectory samples the position in Cartesian space and the orientation as unit quaternions with a sampling rate of 100 Hz. Error trajectories record, how an error occurs starting from an initial position. The last position of the error trajectory is the error position and classifies the 12 different error types with the labelling system mentioned above. Starting from one error position, which defines one type, ten correction trajectories with the correct end position (the peg being in the hole) are recorded through human demonstration. In total, the database contains 120 correction entries and 12 error entries. For the next steps, only the trajectories of the samples are used. Hence the database is made publicly accessible, the remaining data can be used in further research. [10].

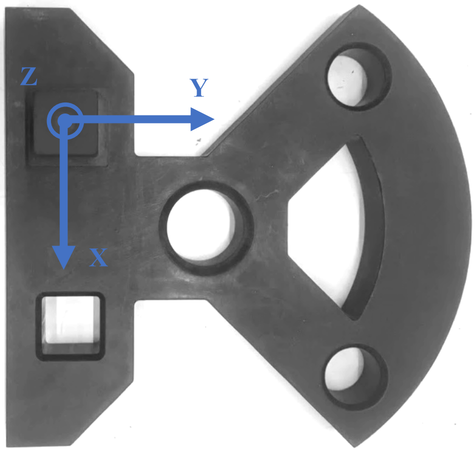

Figure 1: Peg in hole position with the coordinate system for offset (10mm) definition

### 2.2 Averaging

With the aim of representing one error type by its corresponding correction trajectories, a suitable averaging method must be found. The purpose of this is to average the ten correction trajectories of one error type regarding their position and orientation. The resulting average trajectory should represent the correction trajectories of one error type so that it can be used for the similarity measure in the next step. For that reason, two averaging methods are compared. In the first approach, the arithmetic mean is used for the positional components of the trajectory and the algorithm described in [12], to average the quaternions. This method requires a time-independent representation of the trajectories. Another requirement is that each correction trajectory has the same number of sample waypoints, which is not the case for the original data obtained from the database. To fulfil these, each trajectory is first encoded with Gaussian Radial Bases Functions with 25 equidistant kernels [9]. From this interpolation, the same number of 150 waypoints is obtained, which form a time-independent path. Each waypoint has a positional component and a unit quaternion for the orientational component (see Figure 2, dotted path).
As shown in Figure 2, the first averaging approach fails to adequately represent most of the correction trajectories, as there are several modes to solve the error. This is due to the significant changes in the human demonstration that can occur when the robot arm is not always raised in the same way. The resulting average trajectory is in consequence, unable to solve the multiple-modes problem. It is also not representative of one error type since it does not match the positional components of the majority of the corrective trajectories.

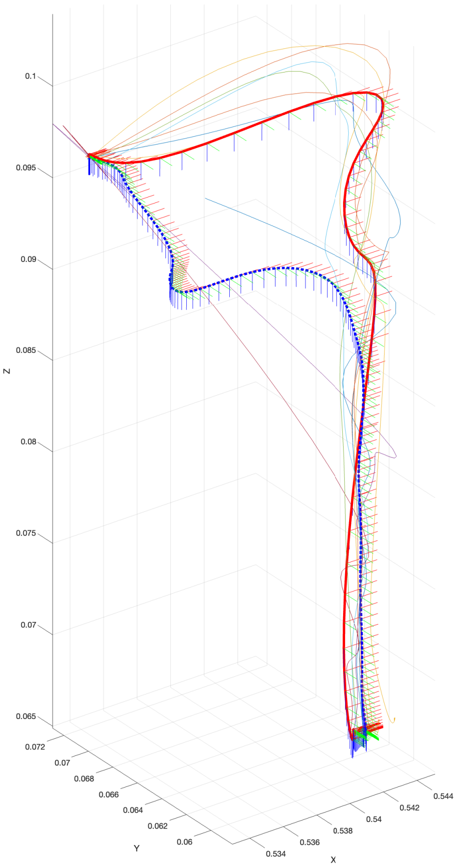


Figure 2: Average trajectory obtained through arithmetic mean (blue/dotted path) and average trajectory obtained through DBA (red/solid path) of one error type with a positional offset of: x = -10mm, y = +10mm and orientation offset around: x = 0°, y = 0°, z = 0°. The thinner paths are the ten corrective trajectories.

Because of that reasons, a second approach is developed, which is based on Dynamical Time Warping Barycenter Averaging [5]. This method uses the Dynamical Time Warping distance in an iterative process. Starting with an arbitrary average trajectory, which is, in this case, one of the correction trajectories, the average trajectory is adapted in every iteration in such a way, that the sum of the squared DTW distances is minimized [5]. Because the resulting positional components differ in their lengths, it is necessary to normalize them to 150 points. This is done by interpolating and back-transforming each of them through RBFs [9], similar to the first approach. Note that, in contrast to the first approach the averaging is done before the normalization through RBFs. The orientational component is achieved like in the first approach and joined with the positional component at the end. A possible improvement is proposed in the discussion. This method has the advantage, that it is time-independent and very robust [5], which means that the mentioned multiple-modes problem can potentially be solved by an averaging method (see Figure 2, solid path).

### 2.3 Similarity Measure and Classification

The second step aims to find a suitable method to classify a newly taught corrective action. This can be done, by comparing the trajectory of a newly taught corrective action with the average trajectory of one class. To do so, it is necessary to define a similarity criterion. As a new approach, the weights of Gaussian Radial Bases Functions seem promising for this purpose. The method has three advantages. First, it is time-independent, second, the orientational component can be included and third, the amount of kernel functions can be set to the same number, through which the weights of the RBFs of different trajectories are comparable [9]. In this case, 25 equidistant kernels are used to calculate 25 weights. For a description of the exact calculation see [9]. Each calculated weight consists of three positional and four orientational components. The positional components were compared to each other by using Eq. (1), where $a_{ji}$ is one positional component of the weight of the average trajectory and $c_{ji}$ is one positional component of one weight of the newly taught corrective action. The Index $i = 1, \dots, 25$ indicates the current weight and $j$ indicates the positional component in Cartesian space.

$$SM_{RBF,p} = \sqrt[2]{\sum_i (a_{xi} - c_{xi})^2 + \left(a_{yi} - c_{yi}\right)^2 + (a_{zi} - c_{zi})^2} \qquad (1)$$

In this way, a similarity measure (SM), based on RBFs can be defined. The lower the SM, the more similar the trajectory of a newly taught corrective action and one average trajectory of one error type. Also, the orientational component can be included by using the angular distance between the two quaternions of one corresponding weight *i* in radians (equation (2)). Here $q_{ai}$ refers to one quaternion component of the weight of the average trajectory and $\overline{q_{ci}}$ refers to the conjugation of the corrective trajectory. By calculating $SM_{RBF\_p} + SM_{RBF\_o}$ the result is able to consider the positional and orientational components of the trajectories.

$$SM_{RBF,o} = \sqrt[2]{\sum_i (2\cos^{-1}(Re(q_{ai} * \overline{q_{ci}})))^2} \qquad (2)$$

To compare this new approach, DTW is used to construct an SM for the positional component only. The method is described in [13] and calculates the DTW distances $(d_x, d_y, d_z)$ in equation (3) instead of the differences of the positional components in equation (1).

$$SM_{DTW} = \sqrt[2]{\sum_i (d_x(a_x, c_x))^2 + \left(d_y(a_y, c_y)\right)^2 + \left(d_y(a_z, c_z)\right)^2} \qquad (3)$$

## 3 Results

To test the reliability, the methods were applied to the database mentioned in section 2.1. In the first step, the average trajectories obtained through the arithmetic mean and DBA are plotted for each of the twelve different error types. One of the twelve error types is plotted in Figure 2. In this step, all ten correction trajectories are used for calculation. By comparing the two resulting average trajectories of one error type with each other, it turns out that the one obtained from DBA represents the correction trajectories qualitatively better. The qualitative

improvement in contrast to the approach, which uses the arithmetic mean is well visible in Figure 2.
In the second step, the two SMs are applied to the database. Therefore, the ten correction trajectories of each error type are split randomly. Seven trajectories of each error type are used to calculate an average trajectory using DBA. The other three samples imitate "newly taught correction trajectories" and are considered correctly classified if the SM combined with the average trajectory of their corresponding error type is the lowest. First, the sum of Eqs. (1) and (2) is used to calculate the SMs for the respective three imitated samples of each error type in combination with the average trajectory of each error type. The accuracy can be calculated by dividing all correctly classified cases by all 36 cases, which results from the three imitated “newly taught correction trajectories” for each of the twelve error types. In the result, the accuracy of the novel method, which uses the weights of RBFs is 72,22%. Compared to that, the accuracy of the DTW distance-based approach, which uses Equation (3) is 100%. If the average is calculated by using the arithmetic mean the results do not differ significantly with an accuracy of 75,00% by using the new RBFs-based method and 97,22% by using the DTW distances-based method. Especially the error type with the configuration: $x = -10mm$, $y = 0mm$; $x = 0°$, $y = 10° \, z = 0°$ is wrongly classified for both averaging approaches by using the RBF-based method. Although there is one significantly deviating correction trajectory, similar to the three lower thin paths in Figure 2, there are no unusual in the raw data that would explain this.

## 4 Discussion

As shown in the results, the new RBF-based method does not provide a reliable SM. It can be thought of modifications of this approach, e.g., to weight $SM_{RBF,p}$ and $SM_{RBF,o}$ differently or to manipulate the number of used kernels. But even then, it is questionable if this approach would succeed.
Since the accuracy of the DTW distance-based approach is the best in combination with the presented DBA method, the focus of further research should lay here. It needs to be mentioned, that these two methods are based on DTW, which could lead to unconsidered effects. The problem description also requires that novel error types should be detected. This can be done by setting a threshold for the SM. The DBA method can be further improved by calculating the orientational component by using the iterative process of [5] with the quaternion conform QDTW distance described in [7]. For the used database, DBA solves the multiple-models problem adequately. However further research is needed to generalize the method. Also a comparison with other approaches, mentioned in [4] could be promising.
Concerning the used database, it is mentionable, that the starting and end position of the correction trajectories is represented by a higher number of points than the middle part (average trajectories in Figure 2), which is caused through the recording configuration. In consequence, this point accumulations contribute more to the SM than the middle part, through which the classification improves.

## 5 Conclusion

It can be concluded that the application of DBA to trajectories qualitatively improves the averaging, tackles the multiple-modes problem and can potentially be used in related works. The combination of the DBA method and the DTW distance-based SM yields the best results for classification. In comparison, the initially promising RBFs-based approach is not reliable. Further research can take advantage of these findings and should focus on the integration of the orientational component.

## Literature


[1] B. Nemec, M. Simonič, and A. Ude, ‘Learning of exception strategies in assembly tasks’, presented at the 2020 IEEE International Conference on Robotics and Automation (ICRA), 2020, pp. 6521–6527.
[2] S. Calinon, ‘Learning from demonstration (programming by demonstration)’, *Encyclopedia of robotics*, pp. 1–8, 2018.
[3] M. Simonič, M. Majcen Hrovat, S. Džeroski, A. Ude, and B. Nemec, ‘Determining Exception Context in Assembly Operations from Multimodal Data’, *Sensors*, vol. 22, no. 20, p. 7962, 2022.
[4] Y. Zhou, J. Gao, and T. Asfour, ‘Learning to Shift Attention for Motion Generation’, 2021, doi: 10.48550/ARXIV.2102.12141.
[5] F. Petitjean, A. Ketterlin, and P. Gançarski, ‘A global averaging method for dynamic time warping, with applications to clustering’, *Pattern Recognition*, vol. 44, no. 3, pp. 678–693, 2011, doi: https://doi.org/10.1016/j.patcog.2010.09.013.
[6] Z. Zhu and H. Hu, ‘Robot Learning from Demonstration in Robotic Assembly: A Survey’, *Robotics*, vol. 7, no. 2, 2018, doi: 10.3390/robotics7020017.
[7] B. Jablonski, ‘Quaternion Dynamic Time Warping’, *IEEE transactions on signal processing*, vol. 60, no. 3, pp. 1174–1183, 2012.
[8] A. Billard, S. Calinon, R. Dillmann, and S. Schaal, ‘Robot programming by demonstration’, in *Springer handbook of robotics*, Springer, 2008, pp. 1371–1394.
[9] L. Žlajpah and T. Petrič, ‘Generation of Smooth Cartesian Paths Using Radial Basis Functions’, in *Advances in Service and Industrial Robotics*, Cham, 2020, pp. 171–180.
[10] Goerz, Ron, Majcen Horvat, Matevz, and Simonič, Mihael, ‘PiH Exception strategies Multimodal Database’. Jan. 25, 2023. doi: 10.5281/ZENODO.7568592.
[11] L. Žlajpah, ‘Robot Blockset’, 2017. repo.ijs.si/leon/robotblockset (accessed Jan. 25, 2023).
[12] F. L. Markley, Y. Cheng, J. L. Crassidis, and Y. Oshman, ‘Averaging Quaternions’, *Journal of Guidance, Control, and Dynamics*, vol. 30, no. 4, pp. 1193–1197, Jul. 2007, doi: 10.2514/1.28949.
[13] H. Sakoe and S. Chiba, ‘Dynamic programming algorithm optimization for spoken word recognition’, *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 26, no. 1, pp. 43–49, 1978, doi: 10.1109/TASSP.1978.1163055.