Deviation tracking with incomplete and distorted data - Application to motion trajectories of industrial robots

Charlotte Lacoquelle, Louise Travé-Massuyès, Xavier Pucel, Nathalie Barbosa Roa, Christophe Merle

Deviation tracking with incomplete and distorted data
Application to motion trajectories of industrial robots

Charlotte Lacoquelle[1,2,3] and **Louise Travé-Massuyès**[1,2] and **Xavier Pucel**[2,4] and **Nathalie Barbosa Roa**[2,3] and **Christophe Merle**[2,3]

[1]LAAS-CNRS, Université Fédérale de Toulouse, CNRS, INSA, Toulouse, France

[2]ANITI, Université Fédérale de Toulouse, France

[3]Vitesco Technologies, Toulouse, France

[4]ONERA, DTIS, Toulouse, France

e-mail: {charlotte, louise}@laas.fr, xavier.pucel@onera.fr, {nathalie.barbosa.roa, christophe.merle}@vitesco.com

Abstract

This paper addresses the problem of detecting faults and wear in robotic arms. The problem is made difficult by the presence of pauses in the robots' routine, and by missing data. We propose a fault detection approach based on a variant of the Dynamic Time Warping algorithm to create a fault indicator insensitive to pauses, delays and missing data, but sensitive to deviations due to faults or wear of the robotic arm's joints. The method is tested on real industrial data from robots that operate in a Printed Circuit Board assembly line.

1 Introduction

Industry 4.0 introduces new technologies in production plants, which allows real-time communication between humans, machines and sensors. This creates new opportunities to reduce downtime, to spot production defects and to improve the supply chain. Robotic arms are used for many tasks and interact with other machines, their operators and other people. There can be several dozens of them deployed in a single production plant. However, each robotic arm can cause the whole production chain to stop when it fails. Vitesco Technologies (VT) produces electronic devices for all types of vehicles, and operates hundreds of robotic arms across dozens of plants. A factory that depends on a large fleet of robots must be able to spot any abnormal execution of robot tasks. Although the robot is equipped with a built-in alarming system, execution monitoring is still necessary to detect small deviations, using a separate system to verify that the robot has succeeded in its task or if its behavior is anomalous [1].

In this paper, we address the problem of anticipating robotic arm failures by detecting abnormal behavior. We assume that the robotic arms repeatedly perform the same pick-and-place task, for which we have nominal data. The behavior of each robot is compared to nominal data in order to detect deviations before the arm fails completely. However, in this approach we face two challenges. First, the nominal data is obtained by simulation, the movements are undisturbed and the sensors' acquisitions, such as scanning a barcode, are instantaneous. Second, the actual data collected from the real robot controller suffers from missing data, which makes it challenging to compare with the simulated behavior.

Our approach consists in creating a fault indicator that relies on Dynamic Time Warping (DTW) to assess the similarity between nominal and performed robot trajectories. The indicator identifies and quantifies the deviation while remaining insensitive to missing data and pauses in the execution. We evaluate our approach on data obtained from a set of Universal Robot (UR) arms deployed in a Vitesco Technologies plant.

This paper is organized as follows. First, we review the state of the art in diagnosis and prognosis of robotic arms in industrial plants. Second, we describe the use case which motivates and on which we evaluate our approach. Third, we formally define our detection approach, based on DTW. Finally, the performance of the approach is highlighted through experimental tests on real execution data from robotic arms.

2 Related work and presentation of the problem

A widespread way to avoid production downtime in a plant is to perform predictive maintenance on its machinery through the estimation of the Remaining Useful Life (RUL) [2].

RUL prediction approaches can be grouped into three categories [3]: statistical model-based, physics model-based and AI. Model-based techniques are commonly used in industrial robotics, but it is difficult to establish an accurate physics model of such a complex mechanical system for prognostics purposes [3]. The AI (data-driven) approaches overcome the need for prior knowledge and fault modes of machinery, but they depend highly on the collection of full life-cycle voluminous and high-quality datasets [4], [5], [6].

In the case where high-quality historical data is not available, it is interesting to study the incipient faults that are not yet critical failures. Indeed, robots are equipped with sensors for position, speed and acceleration, and an internal monitoring system which displays warnings and error codes in case of an event.

Even then, some developing faults might not prevent the operation of the robot and thus be difficult to detect at an early stage, but they might contribute to greater issues [7]. Under such circumstances, the need for a fault indicator, sensible enough to detect developing faults, emerges. The generation of fault indicators in robotics is often addressed with the comparison of real data, collected during robot operation, with predicted behavior from a dynamical model [8], [9], [10], [11]. These approaches take into account nei-

ther the nature of the tasks given to the robot and its operating conditions, nor the controller's monitoring system. This limits the ability to assess the system's health status.

In contrast [12] makes direct use of the repetitive behavior of manipulators to detect deviations in their joint's torque. In this work, data batches are collected from the robot's controller while a program is executed repeatedly and under such conditions that the wear is accelerated. The first batch is supposed to represent the healthy state of the robot, and thus it is used as a reference that is compared to the next batches. An indicator is designed with the kernel density estimates and the Kullback-Leibler divergence. In this work, data batches are associated to exactly one task execution and are previously separated from one another. The authors assume that the time series are directly comparable because there is no distortion in time, the execution pace being regular, and there is no missing data. In our work, these assumptions are not met.

Partitioning a data sequence into subsequences of operating cycles can be seen as a pattern recognition task, i.e., the process of finding a specific subsequence or recognizing a particular shape within a time series. Many techniques have been proposed for this purpose: [13] reduces the dimensions of time series by the Haar Wavelet Transform and indexes the coefficients of the Haar transformed sequences for similarity search. In [14] the pattern, to be found in a search collection, has a waveform shape and is encoded with Markov Models. The authors argue that this approach is more flexible than the distance-based ones. In [15], two series are considered similar if they both fit into an ARIMA model that captures their necessary characteristics. An issue arises when doing pattern mining on signals that suffer from different types of distortions in all axes [16]. In that case, standard similarity (distance) measures are unsuitable because not robust to distortions.

The Dynamic Time Warping (DTW) technique handles well time deformations and speed variations in temporal sequences (Section 3.2.1 of [17]), which is why we chose it in our monitoring approach for robotic arms.

3 Use case: a robotized PCB assembly line

The experimental work is performed on industrial collaborative robots from a Printed Circuit Boards (PCB) manufacturer. They are located on the Back-End lines dedicated to adding connectors and the product housing, after the components have been mounted on the PCB. The robots, 6-axis manipulators, perform various tasks such as emptying or filling containers, stacking, holding a camera for quality control. Their commands are defined in a high-level language directly on the teach pendant of the robot (see Figure 1) by the integration engineers. They specify the target points of the manipulator end-effector in the cartesian space and then the robot control system computes the joint trajectory with inverse kinematics. The positions are monitored with encoders, and a current controller is used to provide a target torque on the motors.

This paper focuses on robots with pick-and-place tasks, especially the ones that unload pallets of components (connectors, PCB, ...) to feed welding or test machines, and might also pick the components from those machines to load back the pallets. The reason behind that choice is that there is a significant number of such robots in the VT plants, and that they execute the same trajectory repetitively. The robot motion is not based on visual guidance to reach the components, instead it follows a fixed trajectory computed using the defined corner points of the pallet. This article studies two robots (A and B) carrying out pick-and-place tasks. Given an operator filled pallet with several components (26 for robot A and 99 for robot B), the robot grasps the components one by one with a gripper. Robot A moves the components from the pallet to a welding machine, and from the welding machine to a test machine. Robot B moves the components from a pallet to a circle conveyor, and closes the conveyor's door after each move.

Figure 1: Picture of a UR robot and its teach pendant

The nature of the robot task determines the shape of the collected output signal: the period of the multivariate time-series corresponds to all the components being handled from the pallet until it is emptied and another pallet is sent; we name this the pallet-cycle. In the signal, a sub-period corresponds to the manipulator's movement over a row of components, this is particularly noticeable in the robot's base joint's angular position.

This pallet-cycle is our reference pattern; it is used to select the data batches from the real data stream that should be compared with a fault-free robot trajectory to generate fault indicators. Given that the precise health state of the robots in the plant is unknown, it is not possible to experiment with a healthy robot as in [12]. Therefore, the nominal trajectory is obtained with a high-fidelity dynamical simulator as described in section 6.1.

3.1 Problem and motivation

A desired joint trajectory $q_d \in \mathbb{R}^6$ is generated by the control system based on the operations in the robot script. They describe the motion Ω of the manipulator's end-effector while it performs the pick-and-place tasks as executed on the production line. The actual joint trajectory $q \in \mathbb{R}^6$ is read by the virtual joint encoders and is collected from the control software into a time-series dataset. This dataset is formed by an ordered sequence for each of the 6 joints, of data lengths $N \in \mathbb{N}$, with a constant sampling period, such

as each time series is in the form:

$$X = [x_1, x_2, ..., x_N] \tag{1}$$

X being the reference trajectory, its length is fixed by the duration of one pallet-cycle performed by the robot, as described in section 3.

The real data consists of the robot signals captured while it follows the same program as the one used on the simulator. These outputs constitute a multivariate time-series data stream from which an ordered sequence Y of length $M \in \mathbb{N}$ is extracted for each joint:

$$Y = [y_1, y_2, ..., y_M] \tag{2}$$

Y is constituted by a stream of pallet-cycles, but, unlike the simulated data, there is no indication about the beginning and the end of the movement over a whole pallet. Besides, the sampling period is supposed to be the same as for X although the time-steps are irregular because of missing data at random moments. The interval of length M has less data points than the sampling should imply. In order to compare the simulation pallet-cycle against real data pallet-cycles, these need to be separated in Y.

Assumption 1 (Full cycle)**.** *We assume that X represents exactly one complete pallet-cycle, and that Y contains at least one complete pallet-cycle.*

Our goal is to produce a fault indicator that detects incipient deviations as soon as possible, in a context of missing data and pauses in the robot cycle execution. To this end we start by finding the subsequence $\mathcal{Y}$ of the sequence Y that best matches the query pattern X, such as:

$$\mathcal{Y} = [y_a, y_{a+1}, ...y_b] \qquad \text{with } 1 \leq a \leq b \leq M \tag{3}$$

Once the best matching $\mathcal{Y}$ subsequence is found, we synthesize an indicator that assesses if the robot trajectory $\mathcal{Y}$, for one pallet-cycle, actually resembles the reference trajectory X.

Requirement 1 (Sensitivity to deviations)**.** *The fault indicator must be sensitive to deviations in the sensor values at some time of the cycle.*

Requirement 2 (Insensitivity to pauses)**.** *The fault indicator must be insensitive to missing data and pauses in the robot movement.*

4 Alignment of distorted time sequences

DTW achieves a non-linear mapping of all the samples between a reference (or a query) and a distorted sequence. It ignores the global and local shifts in the time dimension. DTW is able to compare sequences that have different length without any loss of information, which is not the case with other methods. The output is the remaining cumulative distance between the two sequences.

DTW has been extensively used in the audio processing domain, in particular for classification purposes as in [17], through the use of the overall similarity metric. On time series this metric is used to do fast sequential search on a wide variety of datasets [18]. Conversely, on this paper, the interest lies in the local similarity between each element of the sequences formerly aligned by the optimal warping path.

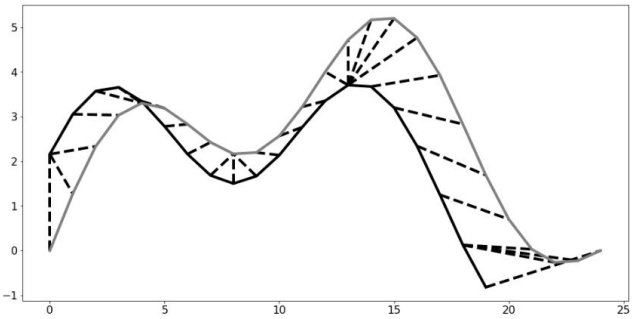

Figure 2: DTW between two signals, with the query pattern X (in black), the sequence Y (in gray) and some warping path pairs (in dashed lines).

4.1 DTW

DTW takes as input a reference time-series X and a time-series Y of respective lengths $N \in \mathbb{N}$ and $\mathrm{M} \in \mathbb{N}$.

$$X = [x_1, x_2, \ldots, x_N] \qquad Y = [y_1, y_2, \ldots, y_M]$$

The output of DTW is a warping path W, that represents the global alignment between the samples of X and Y with respect to the euclidean distance:

$$W = w_1, w_2, \ldots, w_K \tag{4}$$

In a warping path W, each element $w_k = (n, m)$ is a tuple of matching indexes n and m from X and Y. The length K of W is such that $max(N, M) \leq K < N + M$.

All warping paths satisfy the following constraints:

- Both the first and the last elements of the time-series X and Y are aligned to each other: $w_1 = (1, 1)$ and $w_K = (N, M)$.
- Every element of X and Y belongs to at least one alignment w_k.
- The steps of the path in the matrix are monotonically increasing with the step size condition:

$$w_{k+1} - w_k \in \{(1, 0), (0, 1), (1, 1)\} \tag{5}$$

DTW uses the cost matrix $C \in \mathbb{R}^{\mathrm{N*M}}$ computed with a local cost measure. In this paper we use the euclidean distance:

$$C(n, m) = \|x_n, y_m\| \tag{6}$$

The total cost of a warping path W, that represents how well the warped X and Y coincide, is defined as:

$$cost(W) = \sum_{(n,m)\in W} C(n, m) \tag{7}$$

The warping path with minimal cost, denoted P, is the path that starts at cell $(1, 1)$, ends at cell (N, M) of the accumulated cost matrix $D \in \mathbb{R}^{\mathrm{N*M}}$. This matrix is defined as:

$$D(1, m) = \sum_{k=1}^{m} C(1, k) \qquad \text{for } m \in [1..M] \tag{8}$$

$$D(n, 1) = \sum_{k=1}^{n} C(k, 1) \qquad \text{for } n \in [1..N] \tag{9}$$

$$D(n, m) = C(n, m) + \min_{(i,j)\in\Delta} D(n - i, m - j) \tag{10}$$

$$\text{for } \Delta = \{(1, 0), (0, 1), (1, 1)\}, n \in [2..N], m \in [2..M]$$

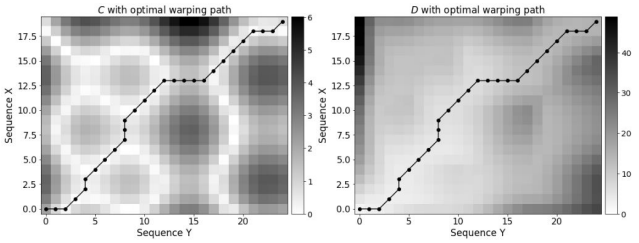

Figure 3: The optimal warping path between two time-series according to DTW is illustrated by black dots through the C (left) and the D (right) matrices.

The global similarity metric is noted and computed as $DTW(X,Y) = cost(P) = D(N,M)$.

In practice, P is computed, according to the original version described by Müller in [17], with an algorithm based on dynamic programming. The content of matrix D is computed backwards through equations (8), (9), and (10) starting at $D(N,M)$ until it reaches $D(1,1)$.

Figure 2 depicts two time series, as well as some pairs of the optimal warping path between them in dashed lines. Figure 3 represents the matrices C and D used to compute this warping path.

4.2 Subsequence-DTW

Subsequence-DTW (S-DTW) was proposed by Müller in [17]. The idea is that the standard DTW can be applied without the constraint that X and Y's boundaries match, i.e. the pattern X can be synchronized with any subsequence $\mathcal{Y}$ in the time-series Y if they happen to match (minimizes the DTW metric to X). This variant of DTW is originally used in audio matching where a short audio clip standing as a query is looked for within an audio database, so that all the corresponding fragments are identified. Based on the properties of DTW, temporal deformations are allowed, so that an audio query can be recognized in another audio piece even if it is played differently.

S-DTW starts, similarly to DTW, by computing the cost matrix C using equation (6). However, the accumulated cost matrix D is defined such as the first row does not accumulate any cost and does not penalize the beginning of X to match with the closest value of Y instead of constraining the endpoints together. Thus, equation (8) is replaced with:

$$D(1,m) = C(1,m) \qquad \text{for } m \in [1..M] \qquad (11)$$

while (9) and (10) remain the same.

Then, the dynamic programming procedure to find the optimal warping path P is applied in the same way as in standard DTW, but the start and stop conditions are modified. Instead of starting at cell $D(N,M)$, it starts at cell $D(N,b)$ such that:

$$b = argmin_{j \in [1:M]} D(N,j) \qquad (12)$$

In addition, instead of reaching the cell $D(1,1)$ as in the standard DTW, S-DTW stops as soon as it reaches a cell of the form $D(1,a)$ for some $a \in [1..M]$. At this point, the subsequence $\mathcal{Y} = [y_a, ..., y_b]$ that optimally matches X has been found.

Figures 4 and 5 illustrate the same example as Figures 2 and 3, except that the S-DTW algorithm is used instead of the DTW one.

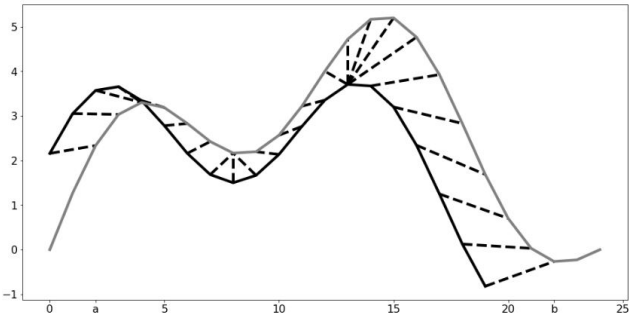

Figure 4: Query pattern X (in black) and sequence Y (in gray) are represented with the alignment given by the optimal warping path between their elements. Note that only a subsequence of Y, denoted by $\mathcal{Y}$ in the text, is matched to X.

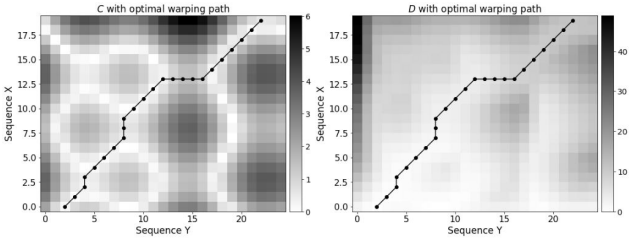

Figure 5: The optimal warping path between two time-series according to S-DTW is illustrated by black dots through the C (left) and D (right) matrix.

5 Deviation tracking in the robot movement trajectories

In this paper, we implement the pattern-recognition technique S-DTW to monitor the execution of robot manipulation tasks. Let us consider a sequence X of reference angular values from a pallet-cycle and a sequence Y of actual angular values. Y is taken of length at least twice as long as X to guaranty that Assumption 1 is fulfilled. By computing S-DTW between X and Y, we are able to:

- find the indexes a and b that characterize a complete pallet-cycle $\mathcal{Y}$ within Y,
- synchronize each element of $\mathcal{Y}$ with the elements of X described by the optimal warping path P,
- detect anomalies thanks to the cost function that can be interpreted as the deviation of the manipulator from its nominal task.

The output of S-DTW is the optimal warping path P described by:

$$P = p_1, p_2, ..., p_L \qquad (13)$$

where

$$max(N,M) \leq L < N + M$$

with $p_1 = (1,a)$ and $p_L = (,b)$ and the middle elements of P are tuples of matching indexes from X and Y:

$$p_l = (n_l, m_l) \in N \times M, l = 2, \ldots, L-1$$

The optimal warping path describes the alignment between a reference pallet-cycle and the one actually performed by a robot. Even if the real signal is distorted, due

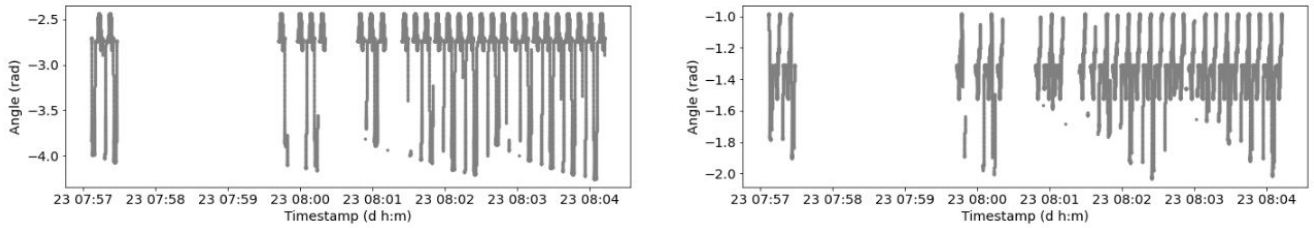

Figure 6: Angular position time-series on two joints (base and shoulder) from an extract of real robot B data.

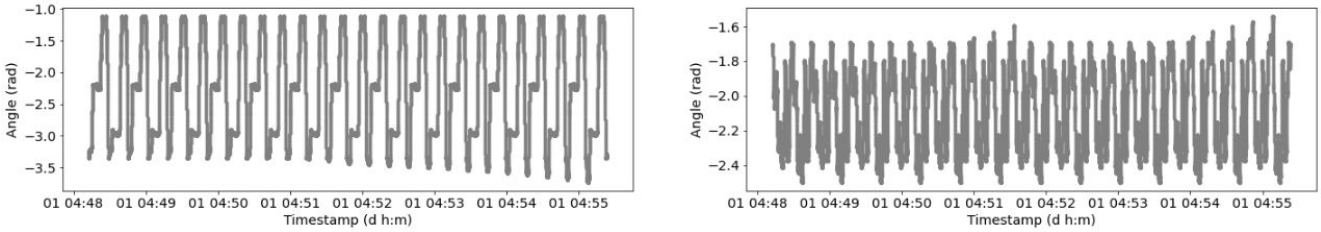

Figure 7: Angular position time-series on two joints (base and shoulder) from simulation of robot A during a pallet-cycle.

to a different production speed compared to the theory, and is suffering from missing data, the alignment is correct.

The synchronization being done, we are able to compare the ideal and the real movements of the robot by defining a local similarity measure obtained for every element of the optimal warping path. This local similarity measure is taken as fault indicator:

$$\mathcal{F}(n_l, m_l) = C(n_l, m_l), l \in [1..L] \tag{14}$$

where $C(n_l, m_l)$ is the cost defined in equation (6). The deviations in the task execution are identified by abnormal points in the signal $\mathcal{F}$.

6 Experimental results

We validate our approach by testing it on experimental data obtained in Vitesco Technologies plants. We first, describe how the reference sequence X and the actual sequence Y are recovered. Then, we evaluate our approach according to the following qualities:

- Sensitivity to pauses in the robot's movement.
- Sensitivity to intervals of missing data.
- Sensitivity to previously identified deviations.
- Scalability (in terms of the X pattern size).

6.1 Data acquisition setup

We collect the data from dozens of robots in real-time, while they perform their daily tasks in the production lines. Each sample includes the timestamp, the angular position of the 6 robot joints, and other variables such as the current and the voltage measures. However only the position measure is used for the sake of this paper. Data acquisition is achieved non-intrusively using a standard TCP/IP connection and the Real-Time Data Exchange (RTDE) interface designed by Universal Robots.

A Python client sets up the variables to collect and the RTDE interface sends the requested output data with a sampling frequency of 125 Hz. For security reasons, the real-time loop in the controller has a higher priority than the RTDE interface, and, accordingly, the controller will skip output packages if it lacks enough computational resources to carry on its tasks. This causes significant gaps at irregular times in our databases, and impacts the diagnosis possibilities. An example of a gap can be seen in sample Table 1 were data collection is interrupted for $136\,ms$. Several interpolation methods have been examined, regrettably, they don't capture complex patterns with these significant gaps. Predicting the missing values has also been considered, however, since the collector is only activated when the robots are moving, the randomness of missing data makes the task challenging. Besides, distinguishing the missing values that should be filled from the actual standstill moments is difficult with the given information. For illustration purposes some irregularly spaced signals are represented in Figure 6.

Observation	Time (s)	Angle (rad) Joint 0	Angle (rad) Joint 1
1	0	-1.737	-1.716
2	0.008	-1.740	-1.714
3	0.016	-1.748	-1.711
4	0.152	-1.801	-1.696
5	0.16	-1.803	-1.695

Table 1: Example of the angular position on joint 0 and 1 collected on a robot with missing values

The reference dataset consists of the sequences defined in (1), obtained by running a simulation of the robot on the manufacturer's software URSim. The output data, joint angles, are also collected via RTDE. In Figure 7 these reference profiles are plotted for two of the virtual robot A joints; they represent a single pallet-cycle while the simulator performs the task given to the real robot. The difference is that the pace is artificial and ideal compared with the actual one from the operating line.

6.2 Experiments

The following section presents the confirmation of the assumptions mentioned in section 3.1 supported by experi-

mental results. Our synchronization and similarity assessment framework was tested on databases acquired from robot A (experiments 1 to 3) and B (experiment 4): the recording of the simulation and the monitoring of the real data as explained in section 3. The computation of S-DTW on two sequences was based on the python package librosa [19], originally designed for audio and music signal processing. The virtual machine which is used for the experiments has 128 GB of RAM.

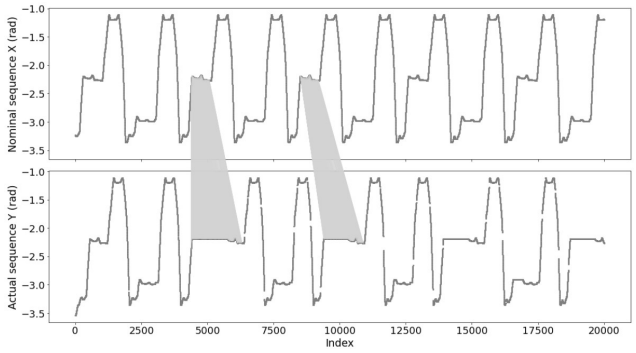

Figure 8: Signals used for Experiment 1. X (above), Y (below), and some S-DTW warping pairs (in gray) during pauses in the robot's movement.

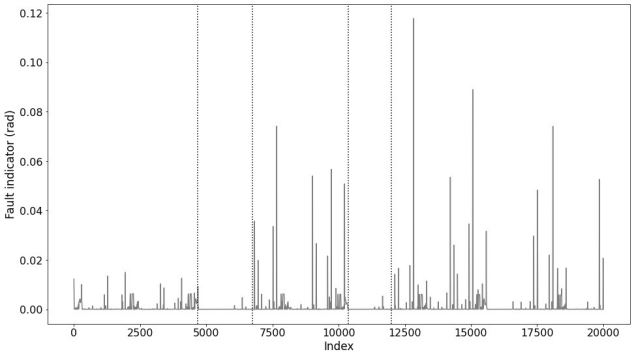

Figure 9: Fault indicator computed for Experiment 1.

Experiment 1 (Sensitivity to pauses in the robot's movement)**.** *In this experiment, we select data sets in which the robot is standstill during a few moments. This is due to a new pallet being unloaded or a machine having to open, or even to the production line being stopped. This causes constant segments in the sequence Y.*

Figures 8 and 9 illustrate how our fault indicator $\mathcal{F}$ behaves in this experiment. Figure 8 shows X on top, and Y below, and draws in gray some of the pairings in the warping path. It shows that the constant segments in Y are correctly associated to their shorter segment in X. Furthermore, during these time windows, Figure 9 shows that the indicator's value is negligible between the corresponding indexes identified by the dots. This means that our indicator is insensitive to pauses in the robot's movement.

Experiment 2 (Sensitivity to missing data)**.** *In this experiment we select Y signals in which there are missing values in the available datasets. This is due to the data collection environment as explained in section 6.1.*

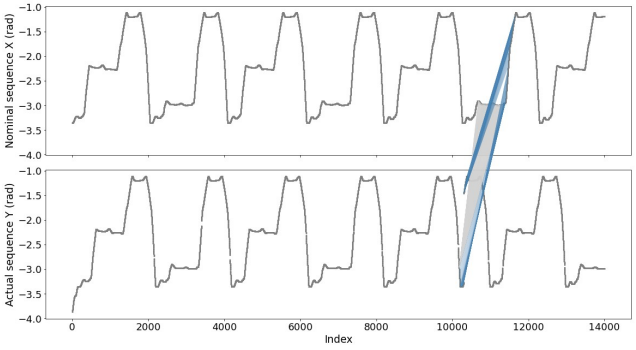

Figure 10: Signals used for Experiment 2. X (above), Y (below), and some S-DTW warping path pairs (in gray and blue).

In Figure 10, reference X and search sequence Y are plotted with lines drawn from an extract of the S-DTW mapping. The gap produced by missing data is rectified by the alignment (in grey) of a long sequence of points from X with a single one from Y. Then, the blue lines show that the algorithm correctly associates reference values with the Y values when they are present. In our case-study, we have systematic missing values and knowledge to characterize the gaps that are induced (as explained in Section 6.1). If we want the fault indicator to be insensitive to these gaps, a threshold can be determined beforehand with normal data including the gaps.

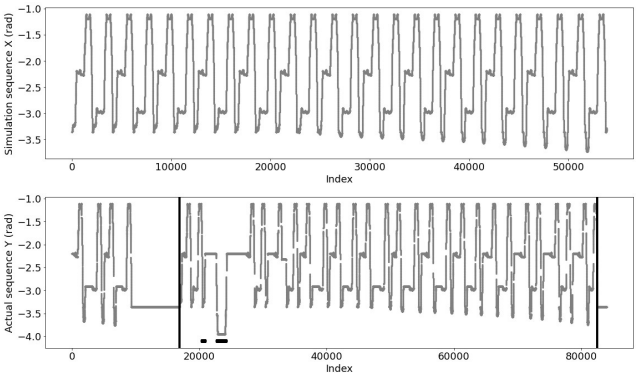

Figure 11: Sequences X (above) and Y (below) used for Experiment 3. The vertical black lines indicate the start and the end of subsequence $\mathcal{Y}$. The segments underlined in black are associated with a high value in the fault indicator.

Experiment 3 (Sensitivity to deviations)**.** *In this experiment we focus on a Y sequence featuring a significant deviation from the reference trajectory due to abnormal behavior from the robot.*

Figure 11 illustrates the reference sequence X and the search sequence Y that were used for this experiment. The collected signal of the real trajectory displays an abnormal behavior. The computation of S-DTW returns the indexes a and b (see Equation (12)) which identify correctly the pallet-cycle, as represented by the vertical black lines on the figure (one somewhere before index 20000 and the other after index 80000). In Figure 12, the fault indicator shows two occurrences of significantly high cost, and the indexes from sequence Y to which they refer are underlined in black in

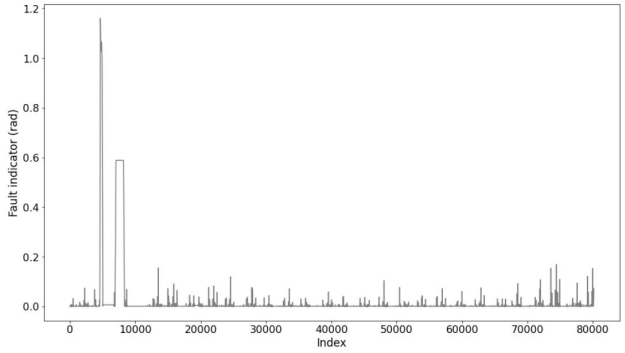

Figure 12: Fault indicator computed for Experiment 3.

Figure 11. This demonstrates that the abnormal behavior has been well pointed out with the proposed approach.

Comparing Figures 12 and 9, we can claim that our fault indicator is significantly more sensitive to the actual deviation of Experiment 3 than the standstills of Experiment 1. Even though in both experiments the indicator value is not null, it is possible to set a threshold, for example 0.2 rad, to separate indicator noise from actual faults. Other factors may generate noise in our indicator: missing data, as in Experiment 2, or differences between the physical robot and its numerical model, but in both cases a threshold is sufficient to distinguish noise from actual deviations.

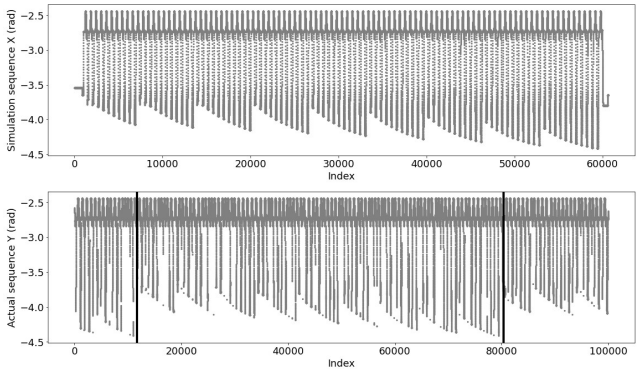

Figure 13: Simulation after downsampling (above) and actual (below) signals of robot B in Experiment 4. The vertical black lines indicate the start and the end of subsequence $\mathcal{Y}$.

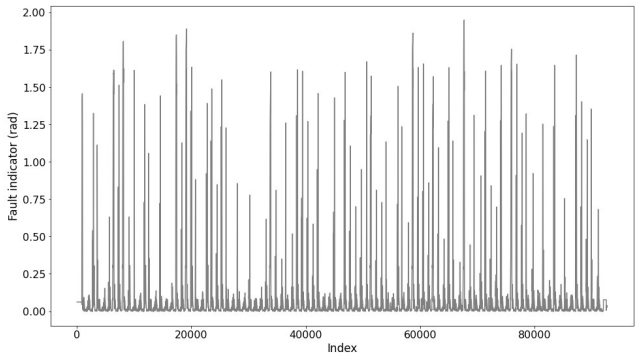

Figure 14: Fault indicator computed for Experiment 4.

Experiment 4 (Scalability)**.** *In this experiment we address a robot with a very long activity cycle, which generates a relatively long reference signal X.*

The datasets from robot A that were used in the first experiments are an optimistic case as they do not have a lot of missing data. As a matter of fact, they were collected outside the collector presented in Section 6.1. Robot B however is connected to this data collector and there is indeed a difference in the quality of the collected data as shown in Figure 13. The pallets unloaded by robot B having 99 components, a pallet-cycle is quite lengthy and the cost matrix becomes too large to be handled directly. To limit the memory consumption on our machine, we downsampled the reference signal by half (62.5 Hz) without losing information. Then, we computed the S-DTW with its classical algorithm version as in Section (4.2). It stands for 16 minutes of recording which is enough for any of our robots' tasks. The algorithm succeeded in finding the indexes that characterize a pallet-cycle (illustrated in Figure 13 by the black lines). We noticed that the result is even more accurate when the inputs X and Y are 6-dimensions matrices with every joint angles. However, a challenge remains: the fault indicator displayed in Figure 14 shows that the gaps induce high costs so extra care is required to differentiate them from real deviations.

6.3 Discussion

As seen with experiment 4, we have memory issues when performing S-DTW if the length of the query is N > 50000 and the length of the search sequence is M > 80000. Indeed, the classic S-DTW algorithm requires storing a matrix of shape (N,M) to compute the optimal alignment between the sequences. In [17], the audio recordings are first preprocessed with a chroma-based feature representation before being compared with DTW, which reduces their size consequently. Attempts have been made to reduce the complexity of DTW: FastDTW [20] is a memory-efficient approximation of it that linearizes the complexity while keeping a good accuracy of the warping path. With its multilevel approach, the warping path is only computed with two columns retained at a time. It is a saving in memory but it implies that the output is only the overall distance between two time-series, and not their index alignment, because the information has been discarded along the algorithm steps. Besides, this contribution and the others such as [21] focus on DTW and not the subsequence variant that we are interested in. We intend to optimize our approach, based on a memory-efficient version of S-DTW [22] which seems promising.

7 Conclusion

A framework based on the S-DTW algorithm is designed to monitor robot motions by assessing the similarity between a reference pattern trajectory for an activity cycle and the real trajectory of the robot during operation. It is based on the alignment of the joint angles time-series. S-DTW makes it possible even in the presence of pauses in the movement and of missing data. The approach synthesizes a fault indicator that identifies and quantifies the deviations from the reference execution. This has been illustrated by a set of experiments performed on real data from a fleet of Universal Robots of the Vitesco Technologies company. The fault indicator is a step torwards predictive maintenance and it can be useful for remote monitoring of the robots. It should

be noted that there is a bias in using the controller angular position outputs as actual data. There can be inherent errors in the angle measurements due to faulty calibration of the encoders. This implies that further work should be done by considering variables that do not depend on calibration matters but are still sensitive to faults.

Acknowledgments

This project is supported by ANITI within the French "Investing for the Future – PIA3" program under the Grant agreement n°ANR-19-PI3A-0004.

References

[1] Daehyung Park, Zackory Erickson, Tapomayukh Bhattacharjee, and Charles C Kemp. Multimodal execution monitoring for anomaly detection during robot manipulation. In *2016 IEEE International Conference on Robotics and Automation (ICRA)*, pages 407–414. IEEE, 2016.

[2] Xiao-Sheng Si, Wenbin Wang, Chang-Hua Hu, and Dong-Hua Zhou. Remaining useful life estimation–a review on the statistical data driven approaches. *European journal of operational research*, 213(1):1–14, 2011.

[3] Yaguo Lei, Naipeng Li, Liang Guo, Ningbo Li, Tao Yan, and Jing Lin. Machinery health prognostics: A systematic review from data acquisition to rul prediction. *Mechanical Systems and Signal Processing*, 104:799–836, 05 2018.

[4] Unai Izagirre, Imanol Andonegui, Aritz Egea, and Urko Zurutuza. A methodology and experimental implementation for industrial robot health assessment via torque signature analysis. *Applied Sciences*, 10(21):7883, 2020.

[5] Cunsong Wang, Ningyun Lu, Senlin Wang, Yuehua Cheng, and Bin Jiang. Dynamic long short-term memory neural-network-based indirect remaining-useful-life prognosis for satellite lithium-ion battery. *Applied Sciences*, 8(11):2078, 2018.

[6] Jinglong Chen, Hongjie Jing, Yuanhong Chang, and Qian Liu. Gated recurrent unit based recurrent neural network for remaining useful life prediction of nonlinear deterioration process. *Reliability Engineering & System Safety*, 185:372–382, 2019.

[7] Rachel Hornung, Holger Urbanek, Julian Klodmann, Christian Osendorfer, and Patrick Van Der Smagt. Model-free robot anomaly detection. In *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 3676–3683. IEEE, 2014.

[8] André Carvalho Bittencourt, Patrik Axelsson, Ylva Jung, and Torgny Brogårdh. Modeling and identification of wear in a robot joint under temperature uncertainties. *IFAC Proceedings Volumes*, 44(1):10293–10299, 2011.

[9] Corbinian Nentwich, Sebastian Junker, and Gunther Reinhart. Data-driven models for fault classification and prediction of industrial robots. *Procedia CIRP*, 93:1055–1060, 2020.

[10] Ikbal Eski, Selcuk Erkaya, Sertaç Savas, and Sahin Yildirim. Fault detection on robot manipulators using artificial neural networks. *Robotics and Computer-Integrated Manufacturing*, 27(1):115–123, 2011.

[11] Benjamin Johnen and Bernd Kuhlenkoetter. A dynamic time warping algorithm for industrial robot motion analysis. In *2016 Annual Conference on Information Science and Systems (CISS)*, pages 18–23. IEEE, 2016.

[12] André Carvalho Bittencourt, Kari Saarinen, and Shiva Sander-Tavallaey. A data-driven method for monitoring systems that operate repetitively-applications to wear monitoring in an industrial robot joint. *IFAC Proceedings Volumes*, 45(20):198–203, 2012.

[13] Kin-Pong Chan and Ada Wai-Chee Fu. Efficient time series matching by wavelets. In *Proceedings 15th International Conference on Data Engineering (Cat. No. 99CB36337)*, pages 126–133. IEEE, 1999.

[14] Xianping Ge and Padhraic Smyth. Deformable markov model templates for time-series pattern matching. In *Proceedings of the sixth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 81–90, 2000.

[15] Konstantinos Kalpakis, Dhiral Gada, and Vasundhara Puttagunta. Distance measures for effective clustering of arima time-series. In *Proceedings 2001 IEEE international conference on data mining*, pages 273–280. IEEE, 2001.

[16] Tomás Kocyan, Jan Martinovic, Pavla Drázdilová, and Katerina Slaninová. Searching time series based on pattern extraction using dynamic time warping. In *DATESO*, pages 129–138. Citeseer, 2013.

[17] Meinard Müller. *Information Retrieval for Music and Motion*. Springer Verlag, 2007.

[18] Thanawin Rakthanmanon, Bilson Campana, Abdullah Mueen, Gustavo Batista, Brandon Westover, Qiang Zhu, Jesin Zakaria, and Eamonn Keogh. Searching and mining trillions of time series subsequences under dynamic time warping. In *Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 262–270, 2012.

[19] Brian McFee, Colin Raffel, Dawen Liang, Daniel P Ellis, Matt McVicar, Eric Battenberg, and Oriol Nieto. librosa: Audio and music signal analysis in python. In *Proceedings of the 14th python in science conference*, volume 8, pages 18–25. Citeseer, 2015.

[20] Stan Salvador and Philip Chan. Toward accurate dynamic time warping in linear time and space. *Intelligent Data Analysis*, 11(5):561–580, 2007.

[21] Thomas Prätzlich, Jonathan Driedger, and Meinard Müller. Memory-restricted multiscale dynamic time warping. In *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 569–573. IEEE, 2016.

[22] Xavier Anguera and Miquel Ferrarons. Memory efficient subsequence dtw for query-by-example spoken term detection. In *2013 IEEE International Conference on Multimedia and Expo (ICME)*, pages 1–6. IEEE, 2013.