

On The Relevance Of Using Temperature For Mechanical Fault Detection In Industrial Robots Based On Dynamic Modeling

Sacha Clavel^a, Mazen Alamir^b and Julien Faure-Favre^a

^aStäubli, 1 Pl. Robert Stäubli, Faverges, 74210, France

^bUniversity of Grenoble Alpes, CNRS, Grenoble INP, GIPSA-lab, 11 Rue des Mathématiques, Grenoble, 38000, France

ARTICLE INFO

Keywords:

Industrial robots
Dynamic model
Deep Learning
Fault Detection
Hybrid Modeling
Temperature variations

ABSTRACT

This paper addresses a key blind spot in industrial robot fault detection and dynamic modeling: the modeling of temperature effects. Although often neglected, ambient temperature and joint temperature significantly influence robot dynamics and vary widely in practice. To address this challenge, a one-class anomaly detection approach is proposed based on an adaptation of a previously introduced hybrid inverse dynamic model combining physics-based and data-driven components, which constructs tight envelopes of normality. The resulting anomaly detector is capable of detecting very slight torque deviations in a real TX2-90 Stäubli 6-axis robot, while remaining robust across a wide trajectory space and continuous joint temperature ranges. The behavior of the envelopes is further analyzed, their effectiveness for anomaly detection on the considered robot is illustrated through preliminary experiments, and the importance of accounting for temperature for mechanical fault detection and localization is highlighted.

1. Introduction

Predictive maintenance (PdM) has gained significant attention as a key paradigm in the era of smart manufacturing. PdM has been enabled by advances in the expanding field of Prognostics and Health Management (PHM), which focuses on the efficient fault detection, diagnosis, and failure prediction in industrial machines. Among these three independent missions, detection is crucial: the earlier the fault is detected, the more time there is to perform a diagnosis, predict its evolution, and prepare for a potential failure. Detection requires identifying the weak signals resulting from deviations of the system's state from its nominal, healthy state.

Industrial arms are key components of modern industry. However, detecting anomalies in such complex systems remains highly challenging in practice, for several reasons:

- **High fault combinatorics:** Industrial arms are multi-joint systems, meaning the number of components and potential faults results in a wide range of possible anomalies.
- **Complex dynamics:** Industrial robots often consist of 4, 6, or even 7 joints, which exhibit highly coupled, intricate, and complex dynamics.
- **Variable operating conditions:** Industrial arms typically operate under highly variable and transient operating conditions, with diverse trajectories, payloads, and at various temperatures.
- **Lack of faulty data:** Due to all the previous causes, the lack of clean and usable failure data from industrial robots is a commonly reported issue [9, 8, 31].

To address this complexity, several data-driven approaches based on Machine Learning (ML) and especially Deep Learning (DL) have been proposed in recent research. Such data-driven approaches have already demonstrated promising results, without the need of prior physical knowledge and time-consuming modeling and identification, as historical physics-based (or *white-box*) methods would require.

In the field of conventional ML, several approaches to anomaly detection in industrial robots rely on combining dimensionality reduction techniques with classifiers, outlier detectors, or regressors. To cite only a few, Principal Component Analysis (PCA) or t-distributed Stochastic Neighbor Embedding (t-SNE) are often coupled with Support Vector Machine (SVM), Gaussian Processes (GP), or Random Forests (RF) [16, 25, 24]. A vast array of other ML approaches has also been explored, from simple Linear Regressions [8], moving to PCA, SVM, or RF used alone [36, 32], as well as Isolation Forests (IF), k-Nearest Neighbors (kNN), k-Means, Local Outlier Factor (LOF), eXtreme Gradient Boosting (XGBoost) [34, 11], among others.

Some approaches lie at the boundary between ML and DL. In particular, Multi-Layer Perceptrons (MLPs), have been widely explored for fault detection in robots [35, 18]. Still at this boundary, some of the ML-oriented approaches attempt to combine ML techniques with DL models. To name just a few, a Generative Adversarial Network (GAN) was combined with SVM and clustering in [31], clustering is combined with a Convolutional Neural Network AutoEncoder (CNN-AE) in [30], and a Convolutional Neural Network (CNN) is combined with PCA and K-Means in [39].

The explosion of DL, driven by advances in computational hardware and data storage and processing, has led to the development of numerous purely DL-based approaches. Among these, methods based on autoencoders (AE) are particularly prevalent, including MLP-AE [41], CNN-AE [9, 19], Long Short Term Memory-AE (LSTM-AE) [38, 5], as well as variants such as Variational AE (VAE) [28, 27], or Sparse AE [15, 22]. Deep learning models are also often used more directly, as in [26] where a residual CNN is used, in [6] a CNN is combined with LSTM, in [42] a CNN is combined with LSTM and attention mechanisms, or in [23] where a CNN is combined with attention, among others.

While most of the aforementioned approaches adopt a full *black-box* formulation, some ML and DL methods incorporate the robot's dynamic model [4, 12]. The underlying intuition is that mechanical failures have a direct impact on the robot's torques, which could be easily spotted by comparison with an accurate nominal dynamic model. This integration offers multiple benefits: it increases explainability, by relying on a tangible robot model, and improves robustness to operating conditions [3].

Indeed, a key limitation of the previously cited approaches is that they are generally designed for specific operating conditions in terms of motion, payload, and especially temperature range. Regarding motions, the vast majority of approaches are meant to be applied to a robot executing repetitive trajectories, such as pick-and-place tasks, and thus might not generalize to unseen motions. However, industrial robots rarely perform the **exact** same motions throughout their lifetime, even in nominally repetitive applications, notably due to production variability. In some cases, they even follow constantly varying trajectories, which should become increasingly common with the advent of collaborative robots.

In order to address this variability, some approaches implement trajectory-change detectors [20, 30], which trigger the refitting of an anomaly detector when a change is detected. These interesting approaches still have shortcomings: they might fail when the motions are constantly varying, and a trajectory change occurring during an ongoing failure could compromise the newly fitted anomaly detector, let alone the commissioning problem of an on-the-fly identified normality of so complicated systems.

Nevertheless, temperature remains the main blind spot in robot anomaly detection and dynamic modeling, certainly because collecting data with temperature changes in a robot is costly, time consuming, and unsafe [13]. Unfortunately, difficult tasks are sometimes still mandatory and this applies to the task of handling temperature variation for two reasons:

- Temperature highly affects the dynamic behavior of robots, especially through its strong impact on friction [14, 7].
- Industrial arms can be used under different ambient temperatures (e.g. cold environments in the food industry, hot environments in the metal industry), and even within a specific single context, the temperature at the joints level can greatly fluctuate (e.g. cold start vs steady state).

Yet, the literature seems to omit temperature effects. To the best of our knowledge, no prior work has integrated temperature into a complete dynamic model of an industrial robot based on ML or DL. Regarding robot fault detection, a few recent studies have considered temperature measurements, but often with limited success. In [34], the authors increase the ambient temperature to simulate a failure mode on a 1-axis hand crafted robot, however, a temperature increase should not necessarily be interpreted as a fault. In [3], joint temperature is collected for condition monitoring of a 6-axis robot, but the authors point out that they were unable

to use it. Similarly, joint temperature features are collected in [29] to detect motion faults in a collaborative robot, but these proved ineffective and might even have led to false alarms, as the natural (i.e. physiological and not necessarily fault-related) aspect of temperature variations was not considered. Finally, [26] proposes a motion-adaptive fault detection method evaluated under 3 ambient temperature contexts (40°C, 50°C and 60°C), with promising results. However, this approach is not validated in a continuous temperature range, and only ambient temperature is considered, neglecting joint-level thermal dynamics, which may evolve independently and significantly affect the robot's state.

In parallel of all this, the use of purely ML or DL based methods for anomaly detection in time-series is increasingly being called into question [2, 33, 1]. Indeed, the characteristics of real-world industrial time-series (e.g. lack of data, poor data quality, varying operating conditions) tend to weaken ML/DL models, which are prone to overfitting and poor generalization. Moreover, explainability is often a critical requirement for industrials, which ML or DL models lack, as they are generally used in a *black-box* manner.

In this contribution, a mechanical fault detection approach is introduced, designed to remain robust across a wide range of operating conditions, in terms of motion and temperature. The method relies on the adaptation of a recently proposed hybrid (or *grey-box*) robot inverse dynamic model, that combines physics-based and data-driven components [10]. Compared to *black-box* models, *grey-box* approaches offer improved generalization capabilities, as well as enhanced interpretability [13, 10], thus addressing key limitations of fully data-driven approaches. The resulting anomaly detector is particularly sensitive while being robust to a broad trajectory space and wide and continuous joint temperature ranges, which to the best of our knowledge, has never been proposed. Furthermore, the model behaves as a one-class classifier, commonly used to detect anomalies in imbalanced datasets (i.e. with very few faulty data), and remains non-invasive by relying solely on the robots' built-in sensors, unlike approaches requiring additional instrumentation [17, 40], which may be impractical in industrial settings.

Note that, as the proposed approach is based on a torque model, it is specifically designed to detect mechanical anomalies that typically affect the robot's torques. It does not target other types of anomalies, such as program-related trajectory anomalies or electronic faults that do not influence torques. Nevertheless, focusing on mechanical failures is particularly relevant, as they account for a major proportion of faults in industrial robots [37, 26].

Based on the previous discussion, the contribution of the present paper is threefold:

- First of all, the paper shows through experimental real-world data that the presumed relevance of including the temperature in the dynamic identified model of a TX2-90 industrial robot is not only theoretically sound but is truly crucial to address the detectability issue. This underlines

the fact that the above identified blind spot in a so wide literature on a so important topic for industry, should be considered and is not an instance of many negligible terms that is legitimate to *pass over* as it is common in many task-oriented modeling. If any, this should be viewed as the main contribution of the present paper.

- The paper also extends a recently proposed *grey-box* dynamic model's structure and demonstrates its effectiveness in capturing the impact of continuously varying temperature on the robot's torques.
- Finally, the paper shows some preliminary results regarding the relevance of using the so-identified temperature-aware dynamic model in the normality characterization and anomaly detection even for quite small deviations from normality.

The paper is organized as follows. Section 2 introduces some notation, recalls the problem formulation, and presents the adaptations made to the hybrid inverse dynamic model proposed in [10]. The concept of *envelopes of normality* is then defined before Section 3 analyzes their behavior, particularly under temperature variations. Section 4 presents their application to anomaly detection using a real Stäubli TX2-90 6-axis robot, in which controller-induced faults as well as actual mechanical faults were deliberately introduced for the sake of illustration.

2. Methodology

After having introduced some notation, this section will formulate the problem, describe the model, and define the metrics used to validate its performances.

2.1. Notation

First, below are listed the main notation of this paper.

FEATURES

The following features/parameters will be used and refer to an industrial arm:

- $n \in \mathbb{R}$ denotes the number of joints of the arm.
- $i \in \{1, \dots, n\}$ denotes a joint index.
- $q, \dot{q}, \ddot{q} \in \mathbb{R}^n$ refer to the joint positions, velocities and accelerations.
- $\tau \in \mathbb{R}^n$ refers to the joint torques.
- $q_d, \dot{q}_d, \ddot{q}_d \in \mathbb{R}^n$ denote specifically the *desired* (i.e. set-point) joint positions, velocities, and accelerations originating from the robot's CPU.
- τ_r denotes specifically the *desired* reference torques required by the joint motors and originating from the robot's PID controller.
- More generally, if τ comes with a subscript such as τ_p, τ_e, τ_c , it always refers to some torques.
- $T_m \in \mathbb{R}^n$ denotes the joint temperatures, *measured* by the temperature probes of the joint motors.
- $t \in \mathbb{R}$ denotes a sampling instant.
- $\delta t \in \mathbb{R}$ denotes the controller sampling time.
- $\mathcal{T}$ denotes a set of sampling instants.

- $\mathcal{T}_t^N$ specifically denotes a sampling time window of length N ending at the sampling instant t :

$$\mathcal{T}_t^N := \{t - (N - 1)\delta t, \dots, t - \delta t, t\} \quad (1)$$

FEATURE ANNOTATION

If $s \in \mathbb{R}^n$ denotes some feature or quantity, the following notation might be used if appropriate:

- $s^{(i)} \in \mathbb{R}$ is the i -th component of s , referring to joint i .
- $s[t]$ denotes the value of s at some sampling instant t .
- $s^{(i)}[t]$ is the feature value for joint i , at some instant t .
- $\tilde{s}$ denotes a time-series of the feature s .
- $\hat{s}$ denotes the output of some model, which, in general, estimates the feature denoted s .
- s_m means that s is a *measured* feature, originating from a sensor.
- s_d means that s is a *desired* (i.e. set-point) feature, originating from the controller.
- Specifically used for temperature, $\bar{T}_m$ denotes the arithmetical mean of the joint temperatures defined as:

$$\bar{T}_m = \frac{1}{n} \sum_i T_m^{(i)} \quad (2)$$

SOME OPERATORS

The following operators will be used:

- If $\mathcal{E} \subset \mathbb{R}$ denotes some set, its indicator function is denoted $\mathbb{1}_{\mathcal{E}}$ such that:

$$\mathbb{1}_{\mathcal{E}}(\tau) = \begin{cases} 1 & \text{if } \tau \in \mathcal{E} \\ 0 & \text{if } \tau \notin \mathcal{E} \end{cases} \quad (3)$$

- If C_i denotes a Boolean condition parameterized by i , the logical OR operator applied over all C_i is denoted $\bigvee_i C_i$.

Additional notation is introduced as needed in the paper.

2.2. Problem Formulation and motivations

Modeling the inverse dynamics of a n -joint robot typically consists in designing a function f :

$$\tau = f(q, \dot{q}, \ddot{q}) \quad (4)$$

that best maps the robot's joint kinematic variables $(q, \dot{q}, \ddot{q}) \in \mathbb{R}^{3n}$ to its joint torques $\tau \in \mathbb{R}^n$. This dynamics is often decomposed as the following sum:

$$\tau = \underbrace{I(q)\ddot{q} + C(q, \dot{q}) + g(q) + F(q, \dot{q}, \ddot{q}, \dots)}_{\tau_p} + \tau_e \quad (5)$$

where $I(q) \in \mathbb{R}^{n \times n}$ is the inertia matrix, C refer to Coriolis and Centrifugal effects and g to gravitation, and F concentrates additional known physics effects (e.g. basic friction, motor inertia, external forces). These first terms are gathered in τ_p that represents all the *easy-to-model* torques so that the last term $\tau_e \in \mathbb{R}^n$ is a residual term which only represents complex and *hard-to-model* physics phenomena (e.g. backlash, joint flexibility, thermal effects).

In previous work [10], a hybrid model M_H was introduced, composed of a physics-based model along with a data-driven correction model, to derive the inverse dynamics of Stäubli industrial robots in a pure open-loop manner:

$$\hat{\tau}_r = M_H(q_d, \dot{q}_d, \ddot{q}_d)$$

where the exact meaning of $q_d, \dot{q}_d, \ddot{q}_d$, and $\hat{\tau}_r$ have been detailed in Section 2.1.

It was demonstrated in [10] that the hybrid nature of M_H allows to achieve high accuracy while maintaining good generalization capabilities compared to a purely data-driven model. However, a remaining limitation lies in its ability to capture certain vibrating phenomena: the model struggles to reproduce oscillatory effects over extended periods, regardless of its size or the amount of training data used.

In order to overcome this limitation, the extension proposed in this paper tries to bound these vibratory effects rather than trying to fully capture them. More precisely:

New approach's paradigm

The objective is to build the so-called *envelopes of normality* $\hat{\mathcal{E}}_k := (\hat{\tau}_k^-, \hat{\tau}_k^+)$ defined by the lower and upper bounds $\hat{\tau}_k^-, \hat{\tau}_k^+ \in \mathbb{R}^n$, around the torques τ_r to be modeled. The meaning of k will be detailed in Section 2.5. These bounds should be such that at any time t and for any joint i , one has the inclusion

$$\tau_r^{(i)}[t] \in [\hat{\tau}_k^{-(i)}[t], \hat{\tau}_k^{+(i)}[t]] \quad (6)$$

satisfied with high probability when the robot operates under *healthy* conditions. Conversely, when an anomaly occurs, the torques τ_r may exceed these bounds with a probability that increases with the severity of the fault.

Obviously, $\hat{\mathcal{E}}_k$ should wrap τ_r as tightly as possible over time. Indeed, the tighter the envelopes, the smaller the amplitudes of the anomalies that can be detected with confidence, as schematized in Fig. 1.

The model used to predict these *envelopes of normality* is a variant of the hybrid model introduced in [10]. This new model, denoted $M_{\mathcal{E}}$, retains its hybrid structure as it is still composed of a physics-based model M_P and a data-driven model M_{θ} . First, the role of M_P is to predict $\hat{\tau}_p$, an estimate of the *easy-to-model* torques τ_p defined in (5), such that:

$$\hat{\tau}_p := M_P(q_d, \dot{q}_d, \ddot{q}_d) \quad (7)$$

To do so, this model is based on several physical sub-models including the Newton-Euler recursive algorithm [21], viscous and dry friction models, and motor shaft inertia models.¹ Then, the goal of M_{θ} is to learn to estimate the residual $\tau_e := \tau_r - \hat{\tau}_p$. In that way, M_{θ} is a residual model so

¹The exact structure of M_P has already been detailed in [10] and will not be recalled in the present paper since it has remained unchanged.

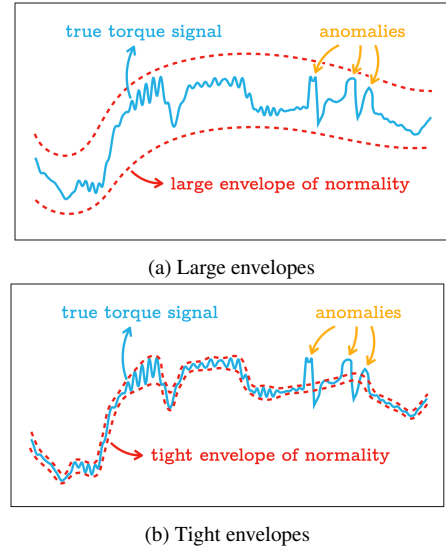


Figure 1: Importance of tight envelopes: Schematic view of measurements under anomalies. (a) Non detection when faulty measurements lie within a wide envelope. (b) Possible detection when the bounds are tighter around the measurements.

that $M_{\mathcal{E}} := M_P + M_{\theta}$ should accurately model τ_r . Its architecture will be presented in the next section.

Finally, a specificity of the approach remains in the fact that $M_{\mathcal{E}}$ is still meant to operate in an open-loop manner:

$$\hat{\mathcal{E}}_k = M_{\mathcal{E}}(q_d, \dot{q}_d, \ddot{q}_d, T_m) \quad (8)$$

meaning that $M_{\mathcal{E}}$ uses the **kinematic set-point** joint positions, velocities, and accelerations together with the joint temperatures, to predict the envelopes $\hat{\mathcal{E}}_k$ that models τ_r . This choice of using the kinematic set-points rather than measurement in the very definition of the identification problem is uncommon, as most comparable approaches systematically incorporate measured positions, velocities, or accelerations into inverse hybrid dynamic models [10]. This deliberate design decision aimed to obtain a model that could be used as a standard feedforward, which notably has the advantage of not being impacted by measurement noise. Thus, $M_{\mathcal{E}}$ captures the dynamics that links the desired trajectories with what really happens in the controller.

2.3. The new residual model M_{θ}

The goal of M_{θ} is to estimate the *hard-to-model* residual $\tau_e := \tau_r - \hat{\tau}_p$ using neural networks. This contribution aims to adapt its architecture so that, on the one hand, M_{θ} allows to bound τ_e rather than attempting to completely capture it, and on the other hand, it incorporates motor joint temperature, which is expected to significantly impact τ_e .

For this purpose, the first step consisted in making M_{θ} model the conditional distribution

$$P(\tau_e | \tilde{q}_d, \tilde{\dot{q}}_d, \tilde{\ddot{q}}_d, \tilde{\tau}_p, \tilde{T}_m)$$

of the current value of τ_e given the time-series $\tilde{q}_d, \tilde{\dot{q}}_d, \tilde{\ddot{q}}_d, \tilde{\tau}_p$, and $\tilde{T}_m$ of length L . These time-series gather the current and some $L - 1$ past values of $q_d, \dot{q}_d, \ddot{q}_d, \tau_p$, and T_m .

Assuming that this n -dimensional distribution can be reduced to n independent normal distributions, this modeling task boils down to estimating their means $\mu_e \in \mathbb{R}^n$ and standard deviations $\sigma_e \in \mathbb{R}^n$. The purpose of M_θ is thus to predict:

$$(\hat{\mu}_e, \hat{\sigma}_e) := M_\theta(\tilde{q}_d, \tilde{\dot{q}}_d, \tilde{\ddot{q}}_d, \tilde{\tau}_p, \tilde{T}_m) \quad (9)$$

To ensure that M_θ learns to predict the appropriate estimates $\hat{\mu}_e$ and $\hat{\sigma}_e$ of μ_e and σ_e , the model is trained using a loss based on the Gaussian log-likelihood. After some simplifications, the log-likelihood of a normal distribution $\mathcal{N}(\mu, \sigma)$ given the measurement y can be expressed as follows.

$$L(\mu, \sigma | y) = -\frac{1}{2} \ln(2\pi) - \ln(\sigma) - \frac{(y - \mu)^2}{2\sigma^2} \quad (10)$$

Parameters μ and σ are optimal when the log-likelihood is maximized, thus, the loss function $\mathcal{L}(\hat{\sigma}_e, \hat{\mu}_e, \tau_e)$ is based on the negative log-likelihood to conform to the loss minimization formalism used in DL, as expressed hereafter.

Training loss of M_θ

The model's loss is defined as the negative log-likelihood averaged over the n robot joints:

$$\mathcal{L}(\hat{\mu}_e, \hat{\sigma}_e, \tau_e) = \frac{1}{2} \ln(2\pi) + \frac{1}{n} \sum_i \left(\ln(\hat{\sigma}_e^{(i)}) + \frac{(\tau_e^{(i)} - \hat{\mu}_e^{(i)})^2}{2\hat{\sigma}_e^{(i)^2}} \right) \quad (11)$$

Minimizing $\mathcal{L}(\hat{\mu}_e, \hat{\sigma}_e, \tau_e)$ over some training set trains M_θ to model $P(\tau_e | \tilde{q}_d, \tilde{\dot{q}}_d, \tilde{\ddot{q}}_d, \tilde{\tau}_p, \tilde{T}_m)$, for each sampling instant and for all joints, with n Gaussian distributions parametrized by $\hat{\mu}_e$ and $\hat{\sigma}_e$.

The final architecture of M_θ is shown in Fig. 2. The network receives input sequences of length $L = 128$ (see [10] for further details about this setting). These time-series are processed through 4 stacked Gated Recurrent Units (GRU)² layers, each with a hidden dimension of 64. The hidden state of the last GRU layer is then fed to two Fully Connected (FC) layers: the first predicts $\hat{\mu}_e$, while the second predicts $\hat{\sigma}_e$ using a Softplus activation that ensures its positivity.

2.4. Learning dataset

In order to train M_θ , a large dataset must be collected, including time-series of $q_d, \dot{q}_d, \ddot{q}_d, \tau_p$, and T_m for model input and τ_e for model output. To ensure the robustness of

²Gated Recurrent Units are a special type of Recurrent Neural Networks (RNN), known for particularly well leveraging temporal correlations in short and long time-series with limited complexity compared to well-known LSTMs. A detailed description has been made in [10].

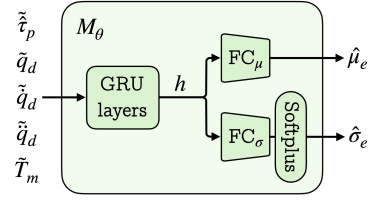


Figure 2: Architecture of M_θ .

the approach across operating conditions, the dataset should span a wide range of kinematic trajectories and temperature.

In order to meet the above conditions, a data collection application was developed to control heating and cooling cycles on the robot. Heating is achieved by executing specific joint-intensive motions until a target temperature is reached. Conversely, cooling is achieved by controlling the robot so that it stands still in a vertical position, thus being free of gravity, until it reaches a desired temperature.

To perform these heating and cooling cycles, the average joint temperature $\bar{T}_m$ (see Section 2.1), is controlled to reach successive target values between 20°C and 55°C with increments of 0.9°C. Each time $\bar{T}_m$ reaches a target temperature, temperature control is briefly paused to trigger a short 15 s data collection, after which temperature control resumes until the next target is reached, and so on. These successive target values of $\bar{T}_m$ will be referred to as *temperature points*, and the data collection program triggered at each *temperature point* will be described in more detail below. Fig. 3 shows the *temperature points* reached by the robot during the first and half cycles of heating and cooling, superimposed with the corresponding joint temperatures $T_m^{(i)}$.

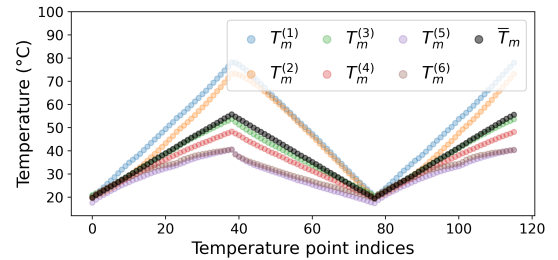


Figure 3: Average joint temperature (black dots) with the corresponding joint temperatures (colored dots) throughout 116 successive *temperature points*. It is recalled that a short data collection is performed at each temperature point.

Each time the robot reaches a *temperature point*, specific data collection motions are triggered and a batch of data is recorded. These motions consist of random point-to-point trajectories, generated from randomly sampled points designed so that the convex hull of the resulting cloud almost covers the entire joint space, along with randomly sampled velocities and accelerations. The randomness is intended to ensure comprehensive coverage of the robot's motions space and thereby improve the model's robustness, in the sense that

Table 1

Dataset description. The velocity and acceleration limits are expressed as a percentage of their maximum values recommended by Stäubli.

Joint position limits	$q_d^{(1)} \in [-180^\circ, 180^\circ]$ $q_d^{(2)} \in [-90^\circ, 90^\circ]$ $q_d^{(3)} \in [-60^\circ, 60^\circ]$ $q_d^{(4)} \in [-180^\circ, 180^\circ]$ $q_d^{(5)} \in [-90^\circ, 90^\circ]$ $q_d^{(6)} \in [-180^\circ, 180^\circ]$
Joint velocity limits	$\dot{q}_d^{(1)}, \dots, \dot{q}_d^{(6)} \in [20\%, 100\%]$
Joint acceleration limits	$\ddot{q}_d^{(1)}, \dots, \ddot{q}_d^{(6)} \in [20\%, 100\%]$
Temperature limits	$\bar{T}_m \in [20^\circ\text{C}, 55^\circ\text{C}]$
Number of trajectories	763
Movements per trajectory	10
Motion type	Joint space
Sampling frequency	250Hz
Sampled signals	$q_d^{(i)}, \dot{q}_d^{(i)}, \ddot{q}_d^{(i)}, \tau_r^{(i)}, T_m^{(i)}$ for $i \in \{1, \dots, 6\}$
Post-calculated signals	$\hat{\tau}_p^{(i)}$ and $\tau_e^{(i)}$ for $i \in \{1, \dots, 6\}$
Total samples	3.45×10^6

future use of the model reduces the need of extrapolation. During these trajectories, q_d , $\dot{q}_d$, $\ddot{q}_d$, τ_r , and T_m are sampled at 250 Hz. The remaining signals are computed offline: $\hat{\tau}_p$ is obtained from $M_P(q_d, \dot{q}_d, \ddot{q}_d)$, and τ_e as the difference $\tau_r - \hat{\tau}_p$. No filtering is applied. Additional details on the collected dataset are provided in Table 1.

The robot used in this study is the 6-axis Stäubli robot shown in Fig. 4. A dataset comprising 10 heating and cooling cycles is collected from this healthy robot, yielding nearly 800 *temperature points* (i.e. 800 random motion cycles), and a total of 3.45×10^6 samples. This dataset, referred to as the *learning set*, is divided into 3 subsets, as described hereafter.



Figure 4: The TX2-90 Stäubli robot used in this study

Working datasets

The *learning set* is split deterministically into 3 working datasets: the first 50% are assigned to a training set of **1 690 382** samples, the next 10% to a validation set of **352 036** samples, and the last 40% to a *threshold-setting* set of **1 407 636** samples whose purpose is detailed in Section 4. The splitting is done in a way that prevents any overlap between the working datasets.

Note that, in addition to the *learning set*, two anomaly-detection datasets were collected separately and are described in Section 4. The first contains controller-induced anomalies and serves to emulate an early deployment phase of the anomaly detector, enabling the final tuning of the detection thresholds. The second one contains real mechanical faults and serves as a final test dataset for the proposed approach.

The training of M_θ is finally performed using mini-batch gradient descent on the loss defined in (11) over the training set for 200 epochs, with the Adam optimizer and default parameters $\beta_1 = 0.9$, $\beta_2 = 0.999$, $\epsilon = 10^{-8}$. The learning rate is exponentially decayed from 10^{-4} to 10^{-5} . The validation loss is monitored at regular intervals for early stopping, and the model achieving the lowest validation loss is considered optimal, a standard practice to mitigate overfitting.

2.5. Construction of the envelopes

Once M_θ is trained, the *envelopes of normality* can be constructed straightforwardly using intervals of the form $\hat{\mu}_e \pm \hat{\sigma}_e$. Indeed, since $\hat{\mu}_e$ and $\hat{\sigma}_e$ should be consistent with Gaussian distributions, the probability that τ_e falls within such intervals should follow the empirical (68-95-99.7) rule, which is experimentally confirmed, provided that the time window over which these probabilities are evaluated is sufficiently large. In particular:

- $\tau_e \in [\hat{\mu}_e - \hat{\sigma}_e, \hat{\mu}_e + \hat{\sigma}_e]$ with 68.27% confidence
- $\tau_e \in [\hat{\mu}_e - 2\hat{\sigma}_e, \hat{\mu}_e + 2\hat{\sigma}_e]$ with 95.45% confidence
- $\tau_e \in [\hat{\mu}_e - 3\hat{\sigma}_e, \hat{\mu}_e + 3\hat{\sigma}_e]$ with 99.73% confidence
- etc.

This leads to the definition of the so-called envelopes.

Envelopes of normality

The *envelopes of normality* $\hat{\mathcal{E}}_k$ are finally defined as

$$\hat{\mathcal{E}}_k := (\hat{\tau}_k^-, \hat{\tau}_k^+) \text{ with } \begin{cases} \hat{\tau}_k^- = \hat{\tau}_p + \hat{\mu}_e - k\hat{\sigma}_e \\ \hat{\tau}_k^+ = \hat{\tau}_p + \hat{\mu}_e + k\hat{\sigma}_e \end{cases} \quad (12)$$

where $k \in \mathbb{R}$ is a parameter that can be tuned to adjust the thickness of the envelopes, thereby affecting the level of confidence they provide.

It can be noted that the width of the envelope of joint i is equal to $2k\hat{\sigma}_e^{(i)}$, thus directly proportional to $\hat{\sigma}_e^{(i)}$. This is why the metric assessing the envelope thickness, defined in the next section, is based on $\hat{\sigma}_e$. Additionally, the center of the envelope, which is the best candidate for defining an estimate of τ_r , will be denoted $\hat{\tau}_r$ and is defined as follows.

$$\hat{\tau}_r = \hat{\tau}_p + \hat{\mu}_e \quad (13)$$

The final architecture of $M_{\mathcal{E}}$ is schematized in Fig. 5. First, the physics-based model M_P predicts $\hat{\tau}_p$ from the kinematic set-point variables. The predicted $\hat{\tau}_p$, together with the kinematic variables and the measured temperatures, is then passed to a buffer that constructs the time-series of current and past values of q_d , $\dot{q}_d$, $\ddot{q}_d$, T_m , and $\hat{\tau}_p$ required by M_{θ} . Based on these inputs, M_{θ} predicts $\hat{\mu}_e$ and $\hat{\sigma}_e$ which define bounds on the residual torque τ_e . The outputs of both models are finally combined to construct the *envelopes of normality* $\hat{\mathcal{E}}_k$ with the desired thickness and level of confidence for each value of the parameter k . Note that the models are multi-joints, taking as input the features of all the joints ($q_d, \dot{q}_d, \ddot{q}_d, \hat{\tau}_p, T_m \in \mathbb{R}^n$) to model all joint torques ($\hat{\tau}_p, \hat{\mu}_e, \hat{\sigma}_e \in \mathbb{R}^n$).

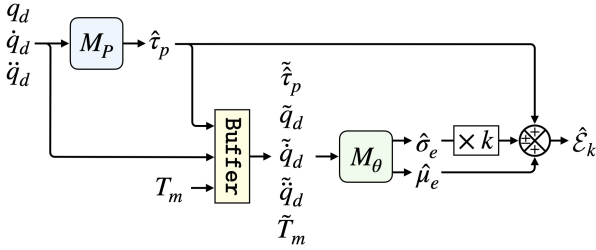


Figure 5: The complete architecture of $M_{\mathcal{E}}$.

2.6. Wrapping metrics

In the next sections, the accuracy of the model is assessed with 3 dimensionless metrics, called *wrapping metrics*, which quantify how well $\hat{\mathcal{E}}_k$ wraps τ_r : the first evaluates $\hat{\mu}_e$, the second evaluates $\hat{\sigma}_e$, and the third evaluates $\hat{\mu}_e$ and $\hat{\sigma}_e$ jointly. These are discussed in the present section.

First, the normalized Mean Absolute Error (nMAE) measures, for each joint i , how accurately the envelope is centered around the torque, relative to the average torque amplitude. More precisely, for $\tau_r^{(i)}$ and $\hat{\tau}_r^{(i)}$ being compared over a set of instants $\mathcal{T}$:

$$\text{nMAE}(\tilde{\tau}_r^{(i)}, \tilde{\tau}_r^{(i)}) := \frac{\sum_{t \in \mathcal{T}} |\tau_r^{(i)}[t] - \hat{\tau}_r^{(i)}[t]|}{\sum_{t \in \mathcal{T}} |\tau_r^{(i)}[t]|} \quad (14)$$

Second, a metric based on the percentiles of $\hat{\sigma}_e$ is used to compare the typical envelope thicknesses relative to the

maximal torque value, during a set of instants $\mathcal{T}$.

$$\mathcal{M}_s(\tilde{\tau}_r^{(i)}, \tilde{\sigma}_e^{(i)}) := \frac{\text{percentile}(\hat{\sigma}_e^{(i)}[t], s)}{\max_{t \in \mathcal{T}} (|\tau_r^{(i)}[t]|)} \quad (15)$$

Finally, the *time-in-envelope* proportion denoted by π_k is used to quantify, for each joint i , the percentage of $\tau_r^{(i)}$ that belongs to its associated envelope $\hat{\mathcal{E}}_k^{(i)} := [\hat{\tau}_k^{-(i)}, \hat{\tau}_k^{+(i)}]$ over a set $\mathcal{T}$ of N instants

$$\pi_k(\tilde{\tau}_r^{(i)}, \tilde{\tau}_r^{(i)}, \tilde{\sigma}_e^{(i)}) := \frac{100}{N} \sum_{t \in \mathcal{T}} \mathbb{1}_{\hat{\mathcal{E}}_k^{(i)}[t]}(\tau_r^{(i)}[t]) \quad (16)$$

where the indicator function $\mathbb{1}_{\hat{\mathcal{E}}_k^{(i)}[t]}$ is defined in section 2.1. In other words, π_k represents the percentage of instants along $\mathcal{T}$ at which the true torque lies within the predicted envelopes, providing a joint evaluation of $\hat{\mu}_e$ and $\hat{\sigma}_e$.

Equivalently, the *time-out-of-envelope* proportion e_k is the complement to 100 of π_k , namely:

$$e_k(\tilde{\tau}_r^{(i)}, \tilde{\tau}_r^{(i)}, \tilde{\sigma}_e^{(i)}) := 100 - \pi_k(\tilde{\tau}_r^{(i)}, \tilde{\tau}_r^{(i)}, \tilde{\sigma}_e^{(i)}) \quad (17)$$

which provides the proportion of $\tau_r^{(i)}$ that deviates from $\hat{\mathcal{E}}_k^{(i)}$ over some set of instants. This is used thereafter as a *naive*³ anomaly detection indicator.

3. Results

In this section, the envelope model behavior is studied and the impact of the temperature feature is notably analyzed.

3.1. Analysis of the envelopes

The envelopes predicted by the model for some of the *threshold-setting* trajectories (see Section 2.4) are shown in Fig. 6. The obtained envelopes clearly succeed in tightly warping the torque signal, especially on the first 3 axes (called major axes). It can be noticed that the torques of the last 3 axes (called minor axes) are much smaller, due to the structural design of industrial arms. Indeed, while the dynamics of the major axes are dominated by gravity and inertia, those of the minor axes are dominated by friction, exhibit a lower signal-to-noise ratio, and are therefore harder to model, resulting in less tightly wrapped envelopes.

To better assess how closely the envelopes wrap the torques, *wrapping metrics* evaluated on some *threshold-setting* trajectories are reported in Table 2. The values for $\mathcal{M}_s$ defined by (15) confirm the previous observation: the envelopes of the major axes exhibit slightly smaller relative thickness, although the envelopes of all axes are still particularly narrow.

³Here e_k is called *naive*, as more sophisticated detection indicators might be developed, for instance based on the distance between τ_r and $\hat{\mathcal{E}}_k$. Exploring such approaches is left for future work, as the main contributions of this paper lie in incorporating temperature effects and providing a first validation of the relevance of normality envelopes for anomaly detection.

Table 2

Some *wrapping metrics* computed across 130 stacked random *threshold-setting* trajectories. The set $\mathcal{T}$ of instants on which the metrics are computed (see Section 2.6) is the union of the sampling instants of the 130 trajectories. The mean and standard deviation observed for 10 independent training runs are given for each metric in order to report the stability of the training process.

	TX2-90 joints					
	1	2	3	4	5	6
$nMAE(\tilde{\tau}_r, \tilde{\tau}_e)$	0.141 ± 0.00031	0.038 ± 0.00021	0.052 ± 0.00032	0.138 ± 0.00053	0.133 ± 0.00271	0.108 ± 0.00126
$\mathcal{M}_{10}(\tilde{\tau}_r, \tilde{\sigma}_e)$	0.011 ± 0.00009	0.007 ± 0.00009	0.010 ± 0.00007	0.018 ± 0.00033	0.012 ± 0.00056	0.017 ± 0.00025
$\mathcal{M}_{50}(\tilde{\tau}_r, \tilde{\sigma}_e)$	0.018 ± 0.00018	0.011 ± 0.00012	0.015 ± 0.00006	0.024 ± 0.00013	0.020 ± 0.00048	0.024 ± 0.00043
$\mathcal{M}_{90}(\tilde{\tau}_r, \tilde{\sigma}_e)$	0.025 ± 0.00022	0.017 ± 0.00013	0.023 ± 0.00019	0.034 ± 0.00031	0.034 ± 0.00050	0.039 ± 0.00089
$\pi_2(\tilde{\tau}_r, \tilde{\tau}_e, \tilde{\sigma}_e)$	94.7 ± 0.199	94.9 ± 0.133	95.0 ± 0.139	95.2 ± 0.088	94.4 ± 0.179	94.5 ± 0.215
$\pi_3(\tilde{\tau}_r, \tilde{\tau}_e, \tilde{\sigma}_e)$	99.2 ± 0.025	99.5 ± 0.018	99.6 ± 0.019	99.5 ± 0.014	99.0 ± 0.040	99.2 ± 0.101

At the same time, the normalized MAE indicates that the model achieves high precision.

Finally, the values of π_k indicate that the predicted $\hat{\mu}_e$ and $\hat{\sigma}_e$ are statistically consistent with the normal empirical rule, reaching 95% with $k = 2$ and 99% with $k = 3$. This supports the validity of modeling τ_e using Gaussian distributions.

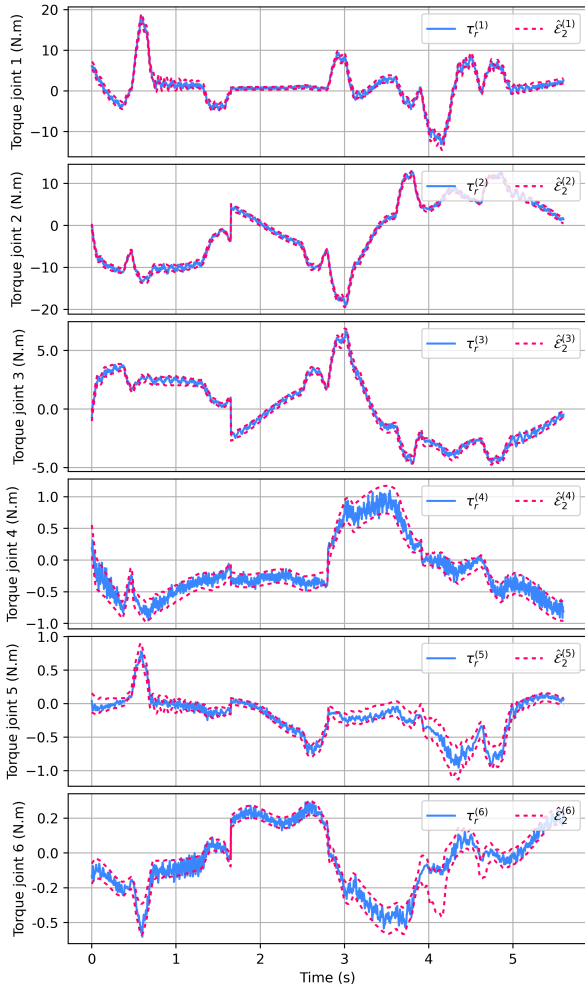


Figure 6: Example of predicted envelopes with $k = 2$ on some *threshold-setting* trajectories.

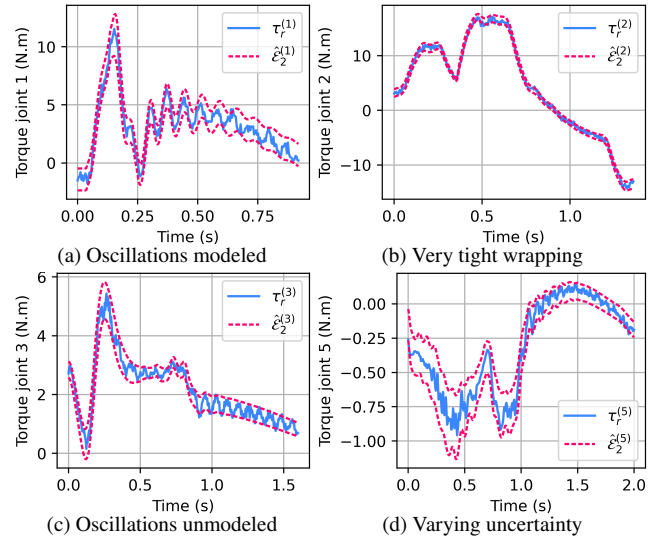


Figure 7: Zooms on the envelopes during the *threshold-setting* trajectories. Note that the time axis are not consistent between the sub-figures, and they are not consistent with Fig. 6 either.

In order to further analyze the envelope behavior, zoomed views of carefully selected segments of the *threshold-setting* trajectories are shown in Fig. 7. First, in Fig. 7b, the envelopes are seen to track the torque particularly tightly on joint 2, including during fast and significant amplitude variations, which is consistent with the metrics reached for this joint in Table 2. A similarly tight wrapping is observed in Fig. 7a, where the model succeeds in capturing a vibratory phenomenon between 0.25 and 0.6, before gradually losing the oscillations. The gradual loss of these vibrations may be attributed to vanishing gradients, which limit the ability of RNNs to capture long-term dependencies. Moreover, in Fig. 7c a similar phenomenon appears at 0.8 s but this time the model fails to capture it and instead produces an envelope that perfectly bounds the oscillations. For this case, the inability of $M_{\mathcal{E}}$ to model vibrations cannot be solely explained by vanishing gradients, as the oscillations are not captured even at their onset. Two hypothesis may explain this behavior: either the relationship between these oscillations

and the model input features is too intricate and complex to be learned, or the oscillations are caused or influenced by external phenomena not included in the input features, making them inherently unobservable by the model.

Regardless of the underlying cause of this behavior along oscillatory intervals, these types of behavior conclusively validate the relevance of using envelope models, and highlight two key characteristics of $M_{\mathcal{E}}$ discussed hereafter.

Relevance of the envelopes

These plots highlight the value of the envelope model:

- When $M_{\mathcal{E}}$ cannot fully capture vibrational effects, it is at least capable of *reasonably tightly* bounding them, which is important for anomaly detection sensitivity.
- $M_{\mathcal{E}}$ models torque uncertainty over time, enabling a context-adaptive envelope thickness as it is features-dependent. The envelopes widen **solely when** uncertainty is high (e.g. due to external phenomena, overly intricate/long-term dependencies, or noise), again crucial for detection sensitivity.

This context-adaptive behavior mentioned above is well illustrated in Fig. 7d: the envelope initially has a reasonable thickness, possibly due to external phenomena present in the training data or overly complex dynamics in this trajectory segment, and then narrows after 1.0 s. It can be seen that the magnitude of the uncertainty can vary rapidly and significantly. This is confirmed by the values of $\mathcal{M}_{10}(\hat{\sigma}_e)$ and $\mathcal{M}_{90}(\hat{\sigma}_e)$ in Table 2, which indicate that typical values of $\hat{\sigma}_e$ can vary by nearly a factor of 3 on all joints.

3.2. Benefits of using T_m

As mentioned in the introduction, a key innovation of $M_{\mathcal{E}}$ revolves in the use of the joint temperatures T_m as part of the input feature vector. In this section, its relevance is assessed by comparing three envelope models:

- $M_{\mathcal{E}}^A$ corresponds to a model that does not use T_m in its input features and that was trained on a second dataset acquired on the robot with stabilized joint temperatures.
- $M_{\mathcal{E}}^B$ does not use T_m as an input feature either, but was trained on the dataset with varying temperatures described in Section 2.4.
- $M_{\mathcal{E}}^C$ uses T_m in input and was trained on the dataset with varying temperatures.

these configurations are summarized in Table 3.

To collect the *learning* set of $M_{\mathcal{E}}^A$, the exact same configuration was used as the one previously described in Table 1, except that the average temperature was stabilized between 37 and 38°C. By so doing, 800 random trajectories were collected, resulting in a total of 3.47×10^6 samples, split into **1 714 049** training samples, **357 825** validation samples and **1 402 060** *threshold-setting* samples.

Table 3

Models configurations.

	Input time-series	Temperature for training
$M_{\mathcal{E}}^A$	$\tilde{q}_d, \tilde{q}_d, \tilde{q}_d, \tilde{\tau}_p$	[37°C, 38°C]
$M_{\mathcal{E}}^B$	$\tilde{q}_d, \tilde{q}_d, \tilde{q}_d, \tilde{\tau}_p$	[20°C, 55°C]
$M_{\mathcal{E}}^C$	$\tilde{q}_d, \tilde{q}_d, \tilde{q}_d, \tilde{\tau}_p, \tilde{T}_m$	[20°C, 55°C]

Table 4

Wrapping metrics of $M_{\mathcal{E}}^A$, $M_{\mathcal{E}}^B$, $M_{\mathcal{E}}^C$ (see Table 3), across 130 stacked random *threshold-setting* trajectories. The set $\mathcal{T}$ of instants on which the metrics are computed (see Section 2.6) is the union of the sampling instants of the 130 trajectories.

		TX2-90 joints					
		1	2	3	4	5	6
$M_{\mathcal{E}}^A$	$\text{nMAE}(\tilde{\tau}_r, \tilde{\tau}_r)$	0.199	0.051	0.059	0.176	0.136	0.135
	$\mathcal{M}_{50}(\tilde{\tau}_r, \tilde{\sigma}_e)$	0.017	0.010	0.013	0.023	0.016	0.023
	$\pi_2(\tilde{\tau}_r, \tilde{\tau}_r, \tilde{\sigma}_e)$	83.0	81.3	88.7	88.5	88.8	88.0
$M_{\mathcal{E}}^B$	$\text{nMAE}(\tilde{\tau}_r, \tilde{\tau}_r)$	0.201	0.047	0.054	0.161	0.138	0.126
	$\mathcal{M}_{50}(\tilde{\tau}_r, \tilde{\sigma}_e)$	0.028	0.014	0.016	0.028	0.018	0.027
	$\pi_2(\tilde{\tau}_r, \tilde{\tau}_r, \tilde{\sigma}_e)$	96.7	96.0	95.4	95.9	92.6	94.3
$M_{\mathcal{E}}^C$	$\text{nMAE}(\tilde{\tau}_r, \tilde{\tau}_r)$	0.142	0.038	0.052	0.139	0.136	0.111
	$\mathcal{M}_{50}(\tilde{\tau}_r, \tilde{\sigma}_e)$	0.018	0.011	0.015	0.024	0.019	0.025
	$\pi_2(\tilde{\tau}_r, \tilde{\tau}_r, \tilde{\sigma}_e)$	94.5	94.8	94.9	95.1	94.2	94.3

The learning models of $M_{\mathcal{E}}^A$, $M_{\mathcal{E}}^B$, and $M_{\mathcal{E}}^C$ used the exact same training settings (in terms of optimizer, epochs, early stopping, etc.). Once trained, the 3 models were evaluated on some of the *threshold-setting* trajectories of the initial dataset described in Section 2.4, corresponding to random motions under varying temperature conditions. The resulting *wrapping metrics* for each model are reported in Table 4.

The fact that $M_{\mathcal{E}}^A$ and $M_{\mathcal{E}}^B$ achieve similar nMAE values, whereas $M_{\mathcal{E}}^C$ is finer, suggests that incorporating temperature enables the generation of envelopes that more accurately track torques under significant thermal variations.

Regarding envelope width, $M_{\mathcal{E}}^A$ predicts slightly tighter envelopes than $M_{\mathcal{E}}^C$, which are in turn tighter than those of $M_{\mathcal{E}}^B$. This is expected: On the one hand, $M_{\mathcal{E}}^A$ predicts the tightest envelopes as it is trained on a limited temperature range, thus with data that have less variability. On the other hand, $M_{\mathcal{E}}^B$ learned to provide wider envelopes, reflecting that torque dynamics is more uncertain when access to temperature is prohibited with training data that include temperature variations.

However, the envelopes of $M_{\mathcal{E}}^A$, although tight, are not consistent for bounding the torque values. Indeed, under normal and healthy operating conditions, π_2 is expected to be close to 95%, which is not the case for $M_{\mathcal{E}}^A$ where it ranges from

81-88% depending on the joint. This inadequate bounding is illustrated in Fig. 8, which compares the models under 3 temperature conditions. When $\bar{T}_m = 37.5^\circ\text{C}$ (i.e. within the training range of M_ε^A), $\hat{\varepsilon}_2^A$ and $\hat{\varepsilon}_2^C$ behave similarly, while $\hat{\varepsilon}_2^B$ is wider. However, at the extreme temperature points, $\hat{\varepsilon}_2^A$ is clearly off the mark, $\hat{\varepsilon}_2^B$ remains wide but still bounds τ , and $\hat{\varepsilon}_2^C$ provides the best compromise.

It becomes clear that **using temperature in the prediction model is essential**. A model trained on a specific temperature range inevitably overfits on this range, and provides tight envelopes that become unusable as soon as temperature fluctuates. Conversely, preventing the model to access to temperature information leads to underfitting: the model is unable to model thermal effects, which increases uncertainty and would make the resulting anomaly detector less sensitive.

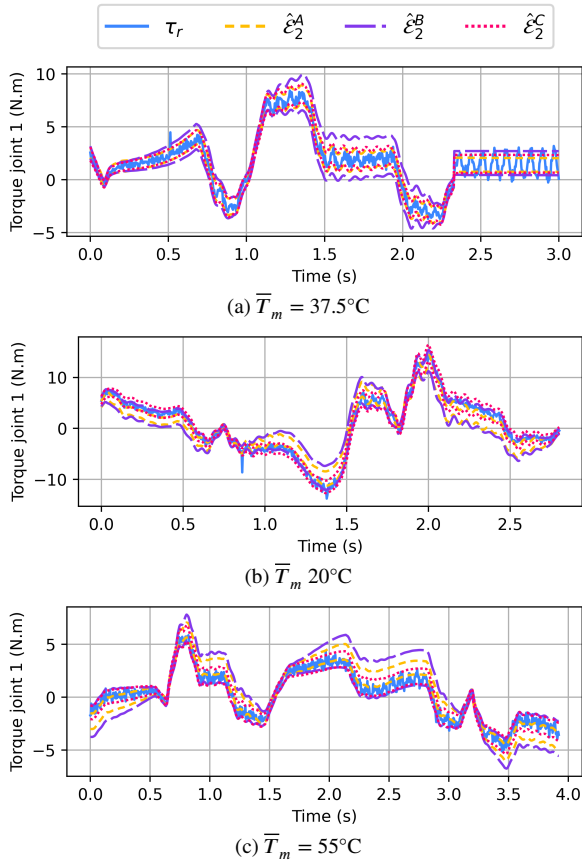


Figure 8: Models behavior at various temperature (see Table 3): (8a): a medium temperature (corresponding to the robot's temperature for the training of M_ε^A) (8b): the lowest temperature, and (8c): the highest temperature.

4. Use of M_ε for anomaly detection

This section demonstrates the relevance of the identified model M_ε for anomaly detection. The methodology for

establishing a detection criterion is presented, and its effectiveness is assessed on synthetic anomalies, corresponding to perturbations deliberately added to the robot's controller.

4.1. Detection criterion

Since $\hat{\varepsilon}_k$ tightly bounds the torque dynamics when the robot operates under *healthy* conditions, these envelopes can be used to detect some even slight deviations from this normality, for example due to component wear or failure. A straightforward way to do so is to monitor e_k , defined in (17), which quantifies the proportion of torque values that lie outside the envelopes over a certain set of instants.

More precisely, the monitored quantity is $e_k[t]$, which in this case corresponds to e_k computed over the time window $\mathcal{T}_t^N$ defined in Section 2.1. In other words, $e_k[t]$ represents the *time-out-of-envelope* proportion computed over a time window of length N , ending at t . In order for $e_k[t]$ to be a relevant anomaly detection indicator, the window length N must be tuned. To illustrate this, Fig. 9 shows the profiles of $e_3[t]$ obtained by sliding $\mathcal{T}_t^N$ along some of the *threshold-setting* trajectories, for three different values of N corresponding respectively to 500 ms ($N = 125$), 5 minutes ($N = 75\,000$), and 30 minutes ($N = 450\,000$).

Because these *healthy* trajectories were not used during training, occasional modeling errors may lead to envelopes that punctually fail to wrap the torques and therefore impacting e_k . This is visible in Fig. 9a, where the 500 ms window leads to highly volatile profiles, with $e_3^{(6)}[t]$ reaching punctually 100%, meaning that the torque of joint 6 entirely exceeds its envelope $\hat{\varepsilon}_3^{(6)}$ during a time window of 500 ms. However, as shown in Fig. 9b and Fig. 9c, increasing the window length smoothens these effects and yields e_3 values that are more statistically consistent with the $\approx 1\%$ expected from the normal empirical rule, further supporting the Gaussian assumption on τ_e .

However, overly large time windows may be inappropriate in an industrial context, as they could mask short-duration anomalies. A window length of about 5 minutes (i.e. $N = 75\,000$) appears to offer a good trade-off: it smooths model inconsistencies and seems affordable in an industrial context. This value is used in the following sections.

Once N is fixed, different values of k could help to differentiate and analyze the severity of the anomalies screened: larger values (e.g. $k = 5$) better discriminate high-amplitude anomalies, while smaller values (e.g. $k = 1$) are more sensitive to subtle ones. In this paper, two values of $k = 1$, and $k = 3$ are considered. However, both k and N might be better tuned depending on the industrial context and the targeted anomalies. This will be addressed in a future paper using several instances of real mechanical failure cases, which are currently being collected.

4.2. Detection thresholds

In order to set the *alarm-firing* thresholds, the quantities $e_k[t]$ with $k \in \{1, 3\}$, defined in the previous section, are

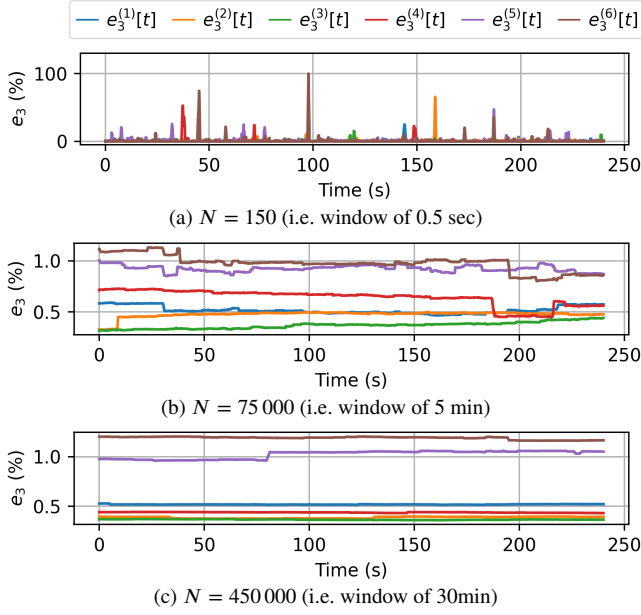


Figure 9: Behavior of $e_3[t]$ for all joints during 4 minutes of the *threshold-setting* trajectories, for different window lengths. Note that these are *healthy* trajectories.

computed along all the *threshold-setting* trajectories (see Section 2.4). Their distribution across the set of instants of these trajectories, denoted $\mathcal{T}_{th}$, is then used to define, for each joint i and $k \in \{1, 3\}$, the thresholds $\eta_{k,s}^{(i)}$, where the parameter $s \in \mathbb{N}$ simply controls the detection sensitivity.

$$\eta_{k,s}^{(i)} = \begin{cases} \text{percentile}(e_k^{(i)}[t], s) & \text{if } s \leq 100 \\ \text{percentile}(e_k^{(i)}[t], 100) \times \frac{s}{100} & \text{otherwise} \end{cases} \quad (18)$$

Note that this threshold definition is relatively simple, yet such quantile-based approaches are widely used in practice for non-parametric *alarm-firing* thresholds, due to their simplicity and low computation time. Besides, defining detection thresholds constitutes itself an active area of research with new approaches that will be considered in future work.

Throughout the robot's operating life, fault detection and localization is made by comparing $e_k^{(i)}[t]$ to its corresponding threshold: when $e_k^{(i)}[t]/\eta_{k,s}^{(i)} > 1$, a fault is supposedly occurring on joint i , otherwise joint i is considered healthy. More specifically, the final *alarm-firing* decision is the worst case over all joints and all values of k :

$$\text{alarm}[t] := \bigvee_{i,k} \left(e_k^{(i)}[t]/\eta_{k,s}^{(i)} > 1 \right) \quad (19)$$

4.3. Generation of subtle controller anomalies

In order to evaluate the anomaly detection framework presented above, and to find relevant values for parameter s of the thresholds $\eta_{k,s}^{(i)}$ introduced in the previous section, an experiment was designed to deliberately generate controller-induced anomalies. More precisely, these anomalies were

induced by injecting excitation signals in real time into the robot's control loop, causing indirect disturbances in the torque signals.

These excitation signals were injected at two points: in the position encoder signal, imitating a position sensor fault (abbreviated *position fault*), and in the current command signal, emulating a current converter fault (abbreviated *current fault*). A highly simplified diagram of Stäubli robot control loops is shown in Fig. 10, indicating the locations of the introduced disturbances.

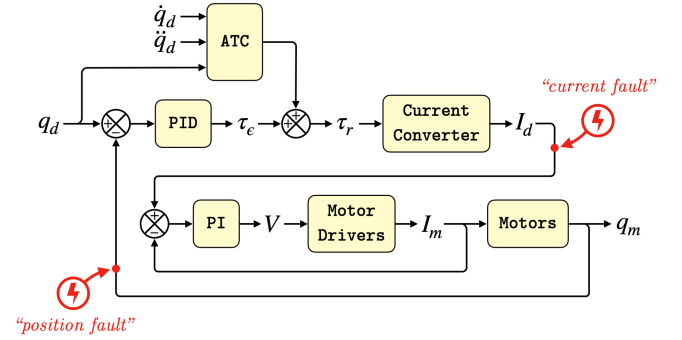


Figure 10: Simplified control loops of Stäubli robot. The red thunderbolts represent the synthetic perturbations/faults that are injected during the experiments.

The injected excitation signals consisted of 10Hz^4 triangular signals of moderate magnitude, chosen for illustration. These perturbations were applied to the robot's second and sixth joints, with three amplitude levels, referred to hereafter as small (S), medium (M) and large (L) respectively.

The corresponding signal amplitudes for each joint are detailed in Table 5. A significant difference can be observed between the disturbance amplitudes applied to the two joints. For position faults, this discrepancy results from joint 6 being approximately 20 times less rigid than joint 2, allowing it to tolerate larger position spikes. For current faults, the difference results from joint 2 being equipped with a motor nearly 3 times more powerful than that of joint 6, enabling it to withstand larger current spikes. Notice that our aim in the process of tuning the amplitudes here and there is to choose amplitudes that clearly emphasize the relevance of using the proposed framework to detect anomaly-induced excursions from normality, while the impact of these faults on the measured signals is not obvious to grasp through simple inspection.

Examples illustrating the magnitude of the corresponding torque disturbances induced by these excitation signals are provided in Fig. 11. For L faults, the impact on torques, while easy to grasp thanks to the colors used, remains quite tricky to detect by means of simple formulation of residuals. For S faults, the impact is barely noticeable and is comparable to

⁴This frequency was chosen to be close to the robot's primary vibration mode, to mimic vibratory excitations that may be induced by certain mechanical faults.

Table 5

Amplitudes of the excitation signals for position and current perturbations which all have 10Hz triangular profile.

Perturbated signal	Severity level	Amplitudes		Unit
		Joint 2	Joint 6	
position	S	0.0243	0.466	mrad
	M	0.109	2.10	
	L	0.218	4.20	
current	S	77.3	26.0	mA
	M	230	78.0	
	L	460	156	

noise. Consequently, these faults constitute a convenient use-case to assess the effectiveness of the proposed framework in detecting subtle anomalies.

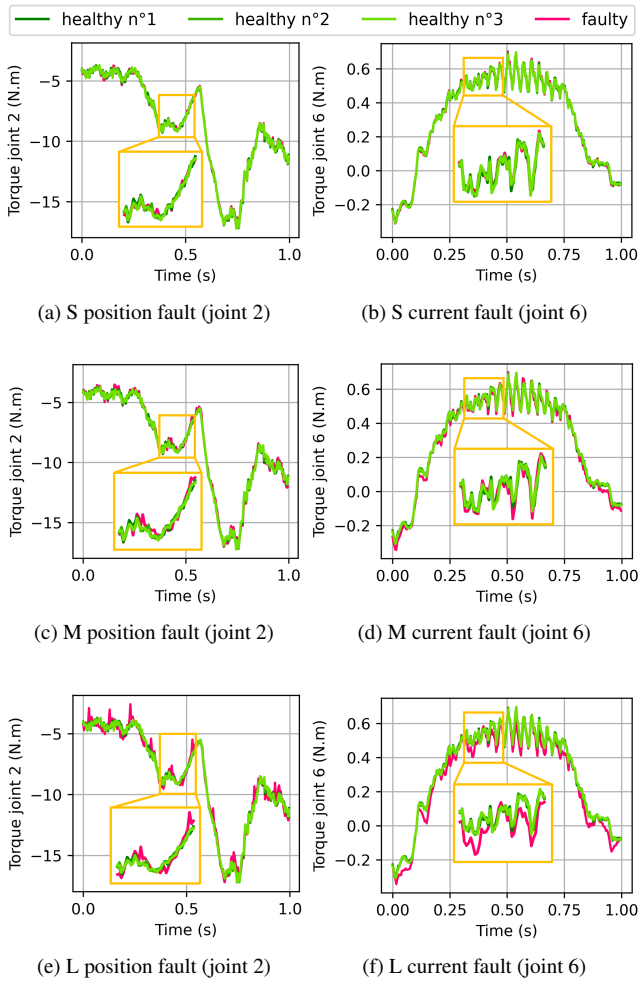


Figure 11: Impact of the faults on the real torque. Here, 4 torque recordings of 1.0 s are superimposed. These recordings were collected from the robot performing the same trajectory with the same joint temperatures: 3 without any perturbation (i.e. healthy) to visualize the ambient noise, and 1 with a perturbation (i.e. faulty).

To construct the anomaly detection dataset, 20 new random trajectories of 5 minutes were collected, including 8 distinct

healthy trajectories and 12 distinct faulty ones. The faulty trajectories correspond to the 12 fault amplitudes listed in Table 5. To assess the robustness of the approach to temperature variations, each trajectory was collected starting from different *temperature points* from $\bar{T}_m = 20^\circ\text{C}$ to $\bar{T}_m = 35^\circ\text{C}$. The trajectories are finally concatenated to enable the computation of the anomaly detection indicators $e_k[t]$. Note that, since these indicators are computed over 5-minute time windows, a 5-minute history is required before they become available. Additionally, since the collected trajectories are approximately 5 minutes long, the first healthy trajectory serves as the initial history buffer. This explains why, in the figures of the next sections, the detection indicators are seen to be unavailable for the first trajectory.

4.4. Detection results for controller anomalies

This section evaluates the relevance of $M_{\mathcal{E}}$ for detecting and localizing the controller-induced torque anomalies, by analyzing the profiles of $e_k[t]$, $k \in \{1, 3\}$, computed over the sliding window $\mathcal{T}_t^N$ along the anomaly detection dataset. In the graphs in this section, a colored background aids interpretation: green for healthy trajectories, and yellow, orange, and red for S, M, and L faults respectively.

First, Fig. 12 shows the obtained $e_3[t]$ profile for the perturbations added to the second joint. This figure allows to validate the two following aspects.

✓ FAULT LOCALIZATION

During M and L perturbations, the indicators successfully identifies the affected joint (joint 2), as $e_3^{(2)}[t]/\eta_{3,110}^{(2)}$ becomes significantly larger than the others. The indicator of joint 3 is occasionally affected, which is expected given the strong dynamic coupling between joints 2 and 3 due to their parallel axes of rotation. It can also be observed that large position faults propagate to all joints, although joints 2 and 3 remain the most impacted.

✓ FAULT SEVERITY ASSESSMENT

The indicators successfully assess the severity of the anomalies, as their values clearly increase with the magnitude of the perturbations.

Nevertheless, it can be seen that the S perturbations only have very little impact on $e_3^{(2)}$, as they have a very small magnitude. This leads to the following point.

✓ TUNING k ALLOWS TO BETTER DISCRIMINATE FAULTS DEPENDING ON THEIR SEVERITY

Indeed, the impact of these subtle perturbations should become more visible with tighter envelopes (i.e. smaller values of k). This is confirmed in Fig. 13 where the profile of $e_1[t]$ is plotted for the same second joint anomalies. Here, the impact on the *time-out-of-envelope* indicator is a little clearer for the S faults, compared to Fig. 13 where it was invisible without the zooms. This supports the idea that the approach can be fine-tuned depending on the type of targeted anomaly, thereby increasing its probability of success.

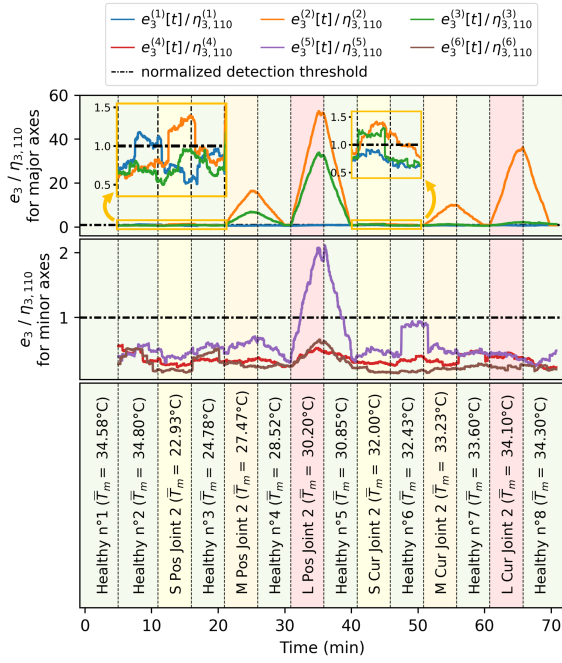


Figure 12: Anomaly detection indicators for faults on joint 2, using $k = 3$ and thresholds with $s = 110$.

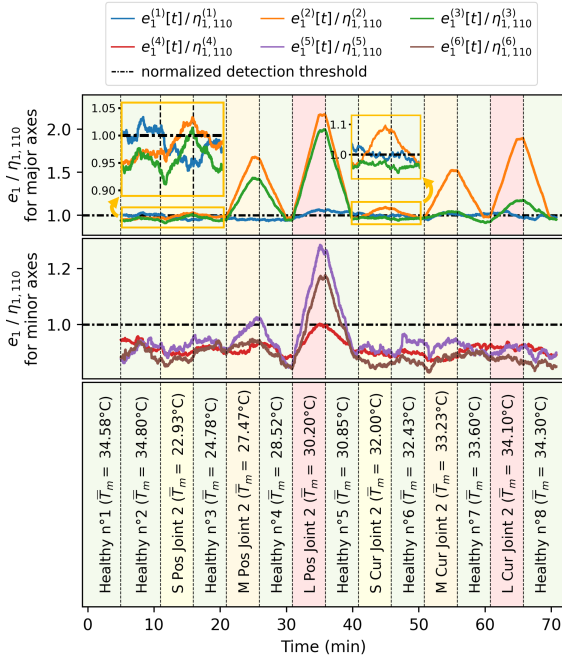


Figure 13: Anomaly detection indicators for faults on joint 2, using $k = 1$ and thresholds with $s = 110$.

✓ ROBUSTNESS TO OPERATING CONDITIONS

Moreover, it can be seen that the anomaly detector exhibits robustness to highly varying operating conditions (i.e. motion randomness and temperature fluctuations), as the behavior of the indicators remains stable despite these variations.

✓ THE USUAL FALSE ALARM/NON DETECTION TRADEOFF IS ADDRESSED BY TUNING s

Finally, the anomaly detection dataset enabled empirical

tuning of the detection thresholds $\eta_{k,s}^{(i)}$. Since these thresholds are derived from healthy data (see Section 4.2), they inevitably require tuning, as is typical during real-world deployment of one-class anomaly detectors. They are tuned via s , which controls detection sensitivity and can be set according to the false alarm rate acceptable to the end user.

First, taking $s = 110$ would enable all successful detection of the 12 anomalies, at the cost of 4 false alarms among the 8 healthy trajectories. These false alarms corresponded to slight exceedances of the indicators during the healthy trajectories, as can be seen in the zooms of Figs. 12 and 13, and thus may be avoided by slightly increasing the thresholds (i.e. increasing s). However, this might also prevent the detection of S anomalies, whose indicators also exceed the thresholds only marginally.

This is confirmed when s is increased to 135. In this case, false alarms are successfully avoided, at the cost of 2 non-detections among the 12 faulty trajectories. These non-detections concern the two smallest faults on joint 6, and are visible in Fig. 14 which shows the profile of $e_1[t]$ along the perturbations added on the sixth joint. This time, the impact of S perturbations, although visible in the indicators, is insufficient to exceed the threshold. Nevertheless, this configuration still enables detection and localization of M and L faults and assessment of their severity.

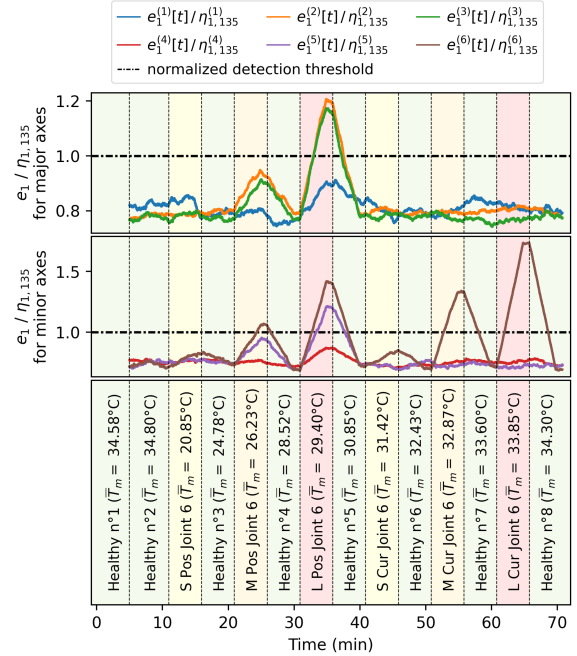


Figure 14: Anomaly detection indicators for faults on joint 6, using $k = 1$ and thresholds with $s = 135$.

These non-detections of the S faults on joint 6 can first be explained by the fact that, as has been remarked in Section 3.1, the envelope of joint 6 is generally wider than that of joint 2. Moreover, it should be noted that these missed detections should not be interpreted as discrediting the model, since the faults in question were voluntarily designed as small as

possible (see Figs. 11a and 11b), precisely to better identify the model's sensitivity limits and to emphasize the trade-off between false alarms and non-detections, which is inherent to all anomaly detection algorithms and unavoidable in real industrial contexts. Besides, in practice, actual mechanical faults can have a much greater impact on torques than that of the S faults considered in this experiment. Consequently, setting $s = 135$ appears to be a good compromise to avoid false alarms while detecting faults of reasonably small amplitudes, and will therefore be used in the next section.

4.5. Detection results for real mechanical faults

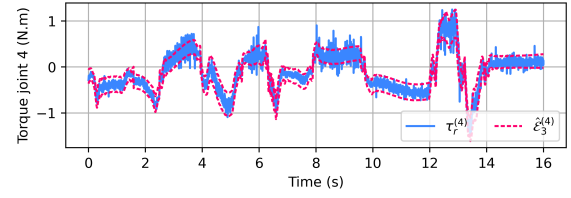
To validate the ability of the approach to detect real mechanical failures, preliminary experiments were conducted, during which healthy parts of the TX2-90 robot were deliberately replaced by components that had known defects of different severities, obtained from experts in the quality and customer service departments of Stäubli. New random trajectories were then collected under varying temperature conditions and used to compare the anomaly detection performances of the models $M_{\mathcal{E}}^A$, $M_{\mathcal{E}}^B$, and $M_{\mathcal{E}}^C$ presented in Section 3.2, in order to emphasize the benefits of incorporating temperature for real mechanical fault detection.

✓ DETECTION OF AN ENCODER FAULT ON JOINT 4

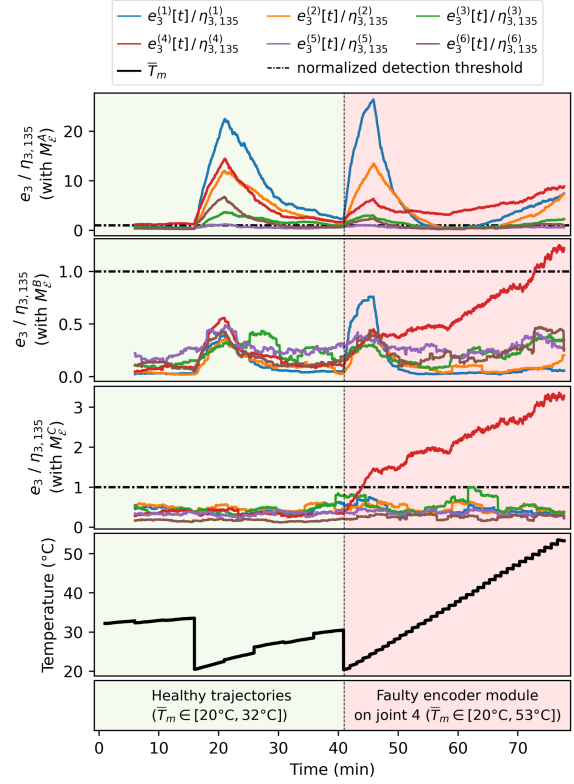
First, the position encoder module of the fourth joint was replaced by one presenting a damaged tooth in its transmission system, corresponding to a fault of moderate severity. The impact induced in the torques can be visualized in Fig. 15a by comparison to the envelopes of $M_{\mathcal{E}}^C$. The anomaly detection indicators obtained with the 3 models are shown in Fig. 15b, where the faulty trajectories have been concatenated with 5-minute long healthy trajectories.

Figure 15 clearly highlights the direct impact of temperature-aware modeling on fault detectability through the following points. First, $M_{\mathcal{E}}^C$ enables the early detection and localization of the fault affecting joint 4, with a progressive increase of $e_3^{(4)}[t]$ which may be consistent with a temperature-dependent behavior of the defect. Second, although $M_{\mathcal{E}}^B$ detects the fault, this model exhibits a significantly delayed response and lower indicator exceedances owing to its wider envelopes. This confirms the reduction in sensitivity discussed in Section 3.2 which may become critical for predictive-maintenance applications where faults must be detected at their earliest stages⁵. Finally, $M_{\mathcal{E}}^A$ would be unusable for anomaly detection because of its strong sensitivity to temperature fluctuations, generating substantial false alarms under changing thermal conditions. In addition, although $M_{\mathcal{E}}^A$ detects deviations during faulty trajectories, the indicators remain consistent only when $\bar{T}_m$ gets close to the 37°C seen during training.

⁵Note that, in this experiment, the severity of the anomaly increases very rapidly, possibly in relation to the temperature increase, which allows $M_{\mathcal{E}}^B$ to detect the fault only 30 minutes after its onset. However, in practice, the deviation from normality due to the progressive wear of some component would be much slower, lasting days if not months. Thus, in real industrial contexts, $M_{\mathcal{E}}^B$ might detect faults significantly later than $M_{\mathcal{E}}^C$.



(a) Deviations from $\hat{\tau}_3^C$ induced by a faulty joint 4 encoder.



(b) Detection indicators.

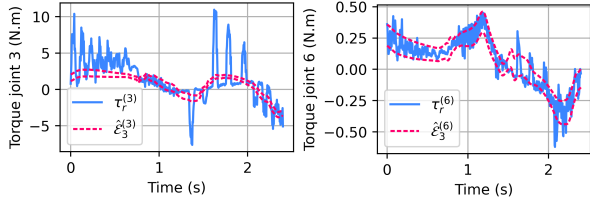
Figure 15: Results obtained with the faulty encoder mounted on joint 4, using $k = 3$ and $s = 135$. (15a) Excerpt of the impacts induced in $\tau_r^{(4)}$. (15b) Anomaly detection indicators obtained for the three models $M_{\mathcal{E}}^A$, $M_{\mathcal{E}}^B$ and $M_{\mathcal{E}}^C$ (see Section 3.2).

✓ DETECTION OF A REDUCER FAULT ON JOINT 3

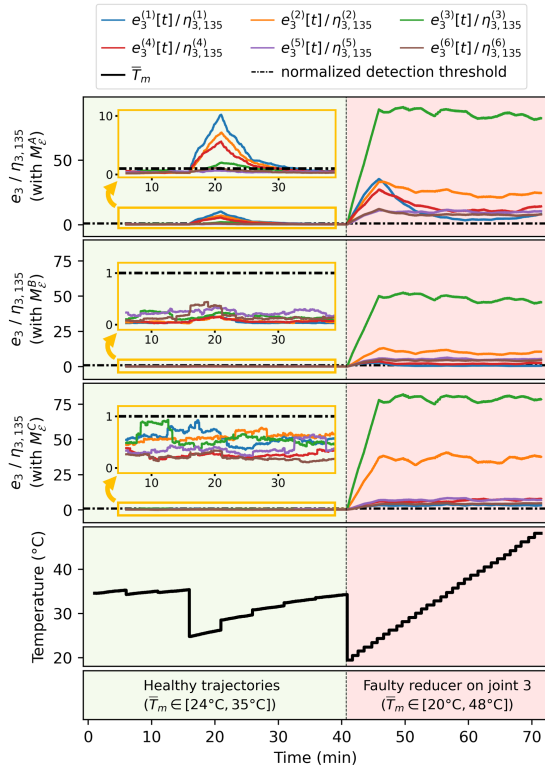
The second experiment was performed with a defect of great severity: the gearbox of the third joint being replaced by one showing important wear, as a result of having been in use for several tens of thousands of hours without maintenance. The corresponding impacts induced in the torques are shown in Figs. 16a and 16b. These plots show that this fault was so severe that its effects propagated up to joint 6. The corresponding detection metrics are shown in Fig. 16.

It can be seen that, this time, the fault is so severe that all the models detect it. This is expected, as even inaccurate detectors would succeed in detecting such unreasonably large anomalies. Nevertheless, important differences remain. $M_{\mathcal{E}}^A$ would still be unsuitable in practice due to the numerous false alarms generated by temperature variations, which

continue to strongly influence its indicators. In contrast, $M_{\mathcal{E}}^B$ and $M_{\mathcal{E}}^C$ exhibit similar behaviors, as expected for such a pronounced anomaly, although $M_{\mathcal{E}}^C$ remains slightly more sensitive, with higher indicator values on joint 3.



(a) Deviations from $\hat{\mathcal{E}}_3$ on joint 3 (b) Spread of the impact to joint 6



(c) Detection indicators.

Figure 16: Results obtained with a faulty gearbox mounted on joint 3, using $k = 3$ and $s = 135$. (16a) Excerpt of the impacts induced in $\tau_r^{(3)}$. (16b) Spread of the impact into $\tau_r^{(6)}$. (16c) Anomaly detection indicators obtained for the three models $M_{\mathcal{E}}^A$, $M_{\mathcal{E}}^B$ and $M_{\mathcal{E}}^C$ (see Section 3.2).

These two preliminary experiments support the relevance of the proposed approach for detecting and localizing actual mechanical failures. In addition, these first results provide encouraging evidence that incorporating temperature improves fault detectability: $M_{\mathcal{E}}^C$ models the torques more accurately, as it can distinguish deviations caused by temperature fluctuations from those caused by mechanical faults. The model is thus more sensitive to small deviations, allowing detection of faults at an early stage, while avoiding false alarms that might be induced by temperature fluctuations.

5. Conclusion and future work

In this contribution, we introduced a novel torque anomaly detection approach designed to remain robust across a broad set of operating conditions. We demonstrated its ability to detect subtle controller anomalies and real mechanical anomalies on a TX2-90 Stäubli robot while maintaining robustness to random motions and, in particular, against a large temperature range. The obtained preliminary results suggest that incorporating temperature improves mechanical-fault detectability and localization, as approaches that do not consider this parameter would not be able to distinguish torque anomalies caused by actual malfunctions from those caused by thermal variations, leading to increased false alarms and reduced sensitivity.

Note that this paper primarily focuses on the methodological aspect of the approach and serves as a basis for future work emphasizing experimental validation with real-world failures. Indeed, future work involves experiments with a large set of real mechanical failures induced on 6-axis and 4-axis robots. The generalizability of the approach to other individuals or models of robots will be studied, as the present study focused solely on a single robot. Additional work could also investigate robustness to payload variations, which appear to be another overlooked factor in anomaly detection for industrial robots.

Finally, the *alarm-firing* threshold remains an active subject of study, as its optimal setting depends on fault types and their statistical characteristics. This trade-off between sensitivity and false alarms is a typical but fundamental challenge for any anomaly detection algorithm and is particularly critical and unavoidable in industrial applications.

References

- [1] Alamir, M., Clavel, S., 2026. Identifying explicit parsimonious piecewise polynomial relationships in industrial time-series: Application to manipulator robots. URL: <https://arxiv.org/abs/2605.28320>.
- [2] Audibert, J., Michiardi, P., Guyard, F., Marti, S., Zuluaga, M.A., 2022. Do deep neural networks contribute to multivariate time series anomaly detection? Pattern Recognition 132. doi:10.1016/j.patcog.2022.108945.
- [3] Ayankoso, S., Gu, F., Louadah, H., Fahham, H., Ball, A., 2024a. Artificial-intelligence-based condition monitoring of industrial collaborative robots: Detecting anomalies and adapting to trajectory changes. Machines 12. doi:10.3390/machines12090630.
- [4] Ayankoso, S., Kaigom, E., Louadah, H., Faham, H., Gu, F., Ball, A., 2024b. A hybrid digital twin scheme for the condition monitoring of industrial collaborative robots. Procedia Computer Science 232, 1099–1108. doi:https://doi.org/10.1016/j.procs.2024.01.108. 5th International Conference on Industry 4.0 and Smart Manufacturing.
- [5] Bayram, B., Duman, T.B., Ince, G., 2021. Real time detection of acoustic anomalies in industrial processes using sequential autoencoders. Expert Systems 38. doi:https://doi.org/10.1111/exsy.12564.
- [6] Bilal, H., Obaidat, M.S., Shamrooz Aslam, M., Zhang, J., Yin, B., Mahmood, K., 2024. Online fault diagnosis of industrial robot using iort and hybrid deep learning techniques: An experimental approach. IEEE Internet of Things Journal 11. doi:10.1109/JIOT.2024.3418352.
- [7] Bittencourt, A.C., Gunnarsson, S., 2012. Static friction in a robot joint—modeling and identification of load and temperature effects. Journal of Dynamic Systems, Measurement, and Control 134. doi:10.1115/1.4006589.

- [8] Borgi, T., Hidri, A., Neef, B., Naceur, M.S., 2017. Data analytics for predictive maintenance of industrial robots, in: 2017 International Conference on Advanced Systems and Electric Technologies (IC-ASET), pp. 412–417. doi:10.1109/ASET.2017.7983729.
- [9] Chen, T., Liu, X., Xia, B., Wang, W., Lai, Y., 2020. Unsupervised anomaly detection of industrial robots using sliding-window convolutional variational autoencoder. IEEE Access 8, 47072–47081. doi:10.1109/ACCESS.2020.2977892.
- [10] Clavel, S., Alamir, M., Faure-Favre, J., 2026. On hybrid inverse dynamic modeling for industrial robots. Mechatronics 115, 103451. doi:https://doi.org/10.1016/j.mechatronics.2025.103451.
- [11] Costa, M.A., Wullt, B., Norrlöf, M., Gunnarsson, S., 2019. Failure detection in robotic arms using statistical modeling, machine learning and hybrid gradient boosting. Measurement 146, 425–436. doi:https://doi.org/10.1016/j.measurement.2019.06.039.
- [12] Fathi, K., van de Venn, H.W., Honegger, M., 2021. Predictive maintenance: An autoencoder anomaly-based approach for a 3 dof delta robot. Sensors 21. doi:10.3390/s21216979.
- [13] Fu, Z., Niu, J., Liu, M., Liu, G., Gao, X., 2026. Dynamics modeling of flexible-joint robotic manipulators: a bibliometric review of physics and ai approaches. Nonlinear Dynamics 114. doi:10.1007/s11071-026-12571-z.
- [14] Gao, L., Yuan, J., Han, Z., Wang, S., Wang, N., 2017. A friction model with velocity, temperature and load torque effects for collaborative industrial robot joints, in: 2017 IEEE/RSJ International Conference on Intelligent Robots and Systems. doi:10.1109/IROS.2017.8206141.
- [15] Hong, Y., Sun, Z., Zou, X., Long, J., 2020. Multi-joint industrial robot fault identification using deep sparse auto-encoder network with attitude data, in: 2020 Prognostics and Health Management Conference (PHM-Besançon). doi:10.1109/PHM-Besançon49106.2020.00036.
- [16] HSU, H.K., TING, H.Y., HUANG, M.B., HUANG, H.P., 2021. Intelligent fault detection, diagnosis and health evaluation for industrial robots. Mechanika 27. doi:https://doi.org/10.5755/j02.mech.24401.
- [17] Jaber, A., Bicker, R., 2017. Wireless fault detection system for an industrial robot based on statistical control chart. International Journal of Electrical and Computer Engineering (IJECE) 7, 3421–3435. doi:10.11591/ijece.v7i6.pp3421-3435.
- [18] Jaber, A.A., Bicker, R., 2016. Industrial robot backlash fault diagnosis based on discrete wavelet transform and artificial neural network. American journal of mechanical engineering 4, 21–31. URL: https://api.semanticscholar.org/CorpusID:39900678.
- [19] Kayan, H., Heartfield, R., Rana, O., Burnap, P., Perera, C., 2024. Casper: Context-aware iot anomaly detection system for industrial robotic arms. ACM Trans. Internet Things 5. doi:10.1145/3670414.
- [20] Kermenov, R., Nabissi, G., Longhi, S., Bonci, A., 2023. Anomaly detection and concept drift adaptation for dynamic systems: A general method with practical implementation using an industrial collaborative robot. Sensors 23. doi:10.3390/s23063260.
- [21] Khalil, W., Dombre, E., 2004. Modeling, Identification and Control of Robots. URL: https://api.semanticscholar.org/CorpusID:1751266.
- [22] Long, J., Mou, J., Zhang, L., Zhang, S., Li, C., 2021. Attitude data-based deep hybrid learning architecture for intelligent fault diagnosis of multi-joint industrial robots. Journal of Manufacturing Systems 61, 736–745. doi:https://doi.org/10.1016/j.jmsy.2020.08.010.
- [23] Lu, K., Chen, C., Wang, T., Cheng, L., Qin, J., 2022. Fault diagnosis of industrial robot based on dual-module attention convolutional neural network. Autonomous Intelligent Systems 2. doi:10.1007/s43684-022-00031-5.
- [24] Narayanan, V., Bobba, R.B., 2018. Learning based anomaly detection for industrial arm applications, in: Proceedings of the 2018 Workshop on Cyber-Physical Systems Security and Privacy, Association for Computing Machinery, p. 13–23. doi:10.1145/3264888.3264894.
- [25] Nentwich, C., Junker, S., Reinhart, G., 2020. Data-driven models for fault classification and prediction of industrial robots. Procedia CIRP 93, 1055–1060. doi:https://doi.org/10.1016/j.procir.2020.04.126. 53rd CIRP Conference on Manufacturing Systems 2020.
- [26] Oh, Y., Kim, Y., Na, K., Youn, B.D., 2022. A deep transferable motion-adaptive fault detection method for industrial robots using a residual-convolutional neural network. ISA Transactions 128, 521–534. doi:https://doi.org/10.1016/j.isatra.2021.11.019.
- [27] Pangione, L., Burroughes, G., Skilton, R., 2021. Variational autoencoder to identify anomalous data in robots. Robotics 10. doi:10.3390/robotics10030093.
- [28] Park, D., Hoshi, Y., Kemp, C.C., 2017. A multimodal anomaly detector for robot-assisted feeding using an lstm-based variational autoencoder. URL: https://arxiv.org/abs/1711.00614.
- [29] Park, Y.S., Yoo, D.Y., Lee, J.W., 2021. Programmable motion-fault detection for a collaborative robot. IEEE Access 9, 133123–133142. doi:10.1109/ACCESS.2021.3114505.
- [30] Polenghi, A., Cattaneo, L., Macchi, M., 2024. A framework for fault detection and diagnostics of articulated collaborative robots based on hybrid series modelling of artificial intelligence algorithms. Journal of Intelligent Manufacturing 35. doi:10.1007/s10845-023-02076-6.
- [31] Pu, Z., Cabrera, D., Bai, Y., Li, C., 2022. A one-class generative adversarial detection framework for multifunctional fault diagnoses. IEEE Transactions on Industrial Electronics 69, 8411–8419. doi:10.1109/TIE.2021.3108719.
- [32] Raouf, I., Lee, H., Kim, H.S., 2022. Mechanical fault detection based on machine learning for robotic rv reducer using electrical current signature analysis: a data-driven approach. Journal of Computational Design and Engineering 9, 417–433. doi:10.1093/jcde/qwac015.
- [33] Rewicki, F., Denzler, J., Niebling, J., 2023. Is it worth it? comparing six deep and classical methods for unsupervised anomaly detection in time series. Applied Sciences 13, 1778. doi:10.3390/app13031778.
- [34] Riazi, M., Zaiane, O., Takeuchi, T., Maltais, A., Günther, J., Lipsett, M., 2019. Detecting the onset of machine failure using anomaly detection methods, in: Ordóñez, C., Song, I.Y., Anderst-Kotsis, G., Tjoa, A.M., Khalil, I. (Eds.), Big Data Analytics and Knowledge Discovery, Springer International Publishing, Cham. pp. 3–12.
- [35] Sabry, A.H., Nordin, F.H., Sabry, A.H., Abidin Ab Kadir, M.Z., 2020. Fault detection and diagnosis of industrial robot based on power consumption modeling. IEEE Transactions on Industrial Electronics 67, 7929–7940. doi:10.1109/TIE.2019.2931511.
- [36] Sathish, V., Ramaswamy, S., Butail, S., 2016. Training data selection criteria for detecting failures in industrial robots. IFAC-PapersOnLine 49, 385–390. doi:https://doi.org/10.1016/j.ifacol.2016.03.084. 4th IFAC Conference on Advances in Control and Optimization of Dynamical Systems ACODS 2016.
- [37] Vallachira, S., Orkisz, M., Norrlöf, M., Butail, S., 2020. Data-driven gearbox failure detection in industrial robots. IEEE Transactions on Industrial Informatics 16, 193–201. doi:10.1109/TII.2019.2912809.
- [38] Wen, X., Chen, H., 2020. Heterogeneous connection and process anomaly detection of industrial robot in intelligent factory. International Journal of Pattern Recognition and Artificial Intelligence 34, 2059041. doi:10.1142/S0218001420590417.
- [39] Xiao, L., Zhao, H., Wu, X., Wang, J., Zhou, Q., 2026. Hybrid learning model for cross-device fault detection of industrial robot joints. Journal of Shanghai Jiaotong University (Science) 31. doi:10.1007/s12204-025-2843-3.
- [40] Yetis, H., Karakose, M., 2018. Image processing based anomaly detection approach for synchronous movements in cyber-physical systems, in: 2018 23rd International Scientific-Professional Conference on Information Technology (IT). doi:10.1109/SPIT.2018.8350461.
- [41] Yun, H., Kim, H., Jeong, Y.H., Jun, M.B.G., 2023. Autoencoder-based anomaly detection of industrial robot arm using stethoscope based internal sound sensor. Journal of Intelligent Manufacturing 34. doi:10.1007/s10845-021-01862-4.
- [42] Zhou, X., Zeng, H., Chen, C., Xiao, H., Xiang, Z., 2022. An attention-enhanced multi-modal deep learning algorithm for robotic compound fault diagnosis. Measurement Science and Technology 34, 014007. doi:10.1088/1361-6501/ac93a5.