

## Full length article

## Learning-based adaptation of robotic friction models

Philipp Scholl<sup>a,\*</sup>, Maged Iskandar<sup>b</sup>, Sebastian Wolf<sup>b</sup>, Jino Lee<sup>b,c</sup>, Aras Bacho<sup>a</sup>,  
Alexander Dietrich<sup>b</sup>, Alin Albu-Schäffer<sup>b,d</sup>, Gitta Kutyniok<sup>a,e</sup>

<sup>a</sup> Ludwig-Maximilians-Universität München, Geschwister-Scholl-Platz 1, Munich, 80539, Bavaria, Germany

<sup>b</sup> German Aerospace Center (DLR), Münchener Straße 20, Weßling, 82234, Bavaria, Germany

<sup>c</sup> Mechanical Engineering Department, KAIST, 291 Daehak-ro, Daejeon, 34141, Yuseong-gu, Republic of Korea

<sup>d</sup> Technical University of Munich, Arcisstraße 21, Munich, 80333, Bavaria, Germany

<sup>e</sup> Munich Center for Machine Learning (MCML), Germany

## ARTICLE INFO

## Keywords:

Robotics

Friction

Data-driven modeling

Neural networks

Transfer learning

External torque estimation

## ABSTRACT

In the Fourth Industrial Revolution, wherein artificial intelligence and the automation of machines occupy a central role, the deployment of robots is indispensable. However, the manufacturing process using robots, especially in collaboration with humans, is highly intricate. In particular, modeling the friction torque in robotic joints is a longstanding problem due to the lack of a good mathematical description. This motivates the usage of data-driven methods in recent works. However, model-based and data-driven models often exhibit limitations in their ability to generalize beyond the specific dynamics they were trained on, as we demonstrate in this paper. To address this challenge, we introduce a novel approach based on residual learning, which aims to adapt an existing friction model to new dynamics using as little data as possible. We validate our approach by training a base neural network on a symmetric friction data set to learn an accurate relation between the velocity and the friction torque. Subsequently, to adapt to more complex asymmetric settings, we train a second network on a small dataset, focusing on predicting the residual of the initial network's output. By combining the output of both networks in a suitable manner, our proposed estimator outperforms the conventional model-based approach, an extended LuGre model, and the base neural network significantly. Furthermore, we evaluate our method on trajectories involving external loads and still observe a substantial improvement, approximately 60%–70%, over the conventional approach. Our method does not rely on data with external load during training, eliminating the need for external torque sensors. This demonstrates the generalization capability of our approach, even with a small amount of data – less than a minute – enabling adaptation to diverse scenarios based on prior knowledge about friction in different settings.

## 1. Introduction

In the context of advancing human–robot collaboration in manufacturing [1–4], where physical interaction between humans and robots is becoming increasingly important, the accurate estimation of interaction forces is critical for ensuring safety [5–7]. Achieving this often involves the precise estimation of external forces, which can be accomplished by mounting force–torque sensors on the robot's end-effector. Unfortunately, many industrial robots do not have these sensors due to their high cost [8]. This situation emphasizes the need for a reliable, cost-effective, and sensorless method for estimating external torque, which, in turn, necessitates the development of an accurate friction model.

A precise model of friction [9] has the potential to enhance the functionality of robotic systems in multiple aspects. It plays a crucial role in achieving energy-efficient computations, enhancing the precision of dynamic simulations, improving control performance at the joint level [10–13], or detecting slip [14]. To mitigate the undesirable

effects of friction, a model-based friction compensation approach is commonly integrated into the control system. This approach finds widespread applications across multiple domains, including joint-level control [15,16], safe human–robot interaction [17–20], and the enhancement of external torque estimation precision during interactions with the environment. Furthermore, friction is an important factor in the degradation curves of the components of a robot, thus, a precise friction estimation also improves the predictive maintenance [21,22]. However, modeling the friction is highly complicated since it depends on a multitude of factors limiting the applicability of model-based approaches [23–28].

## 1.1. Related work

The success of neural networks (NN) in the last decade [29–32] inspired engineers to explore their application in modeling complex

\* Corresponding author.

E-mail address: [scholl@math.lmu.de](mailto:scholl@math.lmu.de) (P. Scholl).

**Table 1**  
Overview of data-driven approaches to model the friction and their application.

Approach	Application goal	Method
Selmic and Lewis [33]	Friction estimation	Fully connected NN with discontinuous activation functions
Ciliz and Tomizuka [34,35]	Motion control	Hybrid of a parametric model and NN
Huang and Tan [36]	Motion control	Combination of two NNs
Guo et al. [37]	Motion control	Fully connected NN with discontinuous activation functions
Hirose and Tajima [38]	Friction estimation	Long Short-Term Memory (LSTM) network
Tu et al. [39]	Friction estimation	NN initialized using Genetic Programming (GP)
Liu et al. [40]	External torque estimation	NN with sigmoid activation

dynamics, including friction. An overview of data-driven approaches for friction modeling is given in Table 1. Selmic and Lewis [33], for example, developed neural network models for friction that utilized specialized architectures with discontinuous activation functions. This design aimed to enhance the fitting of the observed friction data, ultimately reducing the number of required neurons, training time, and data. Another common approach is to use a hybrid method consisting of a simple parametric model and a complicated neural network. Ciliz and Tomizuka [34,35] showed that this approach improves over both, a single parametric model and a single neural network, since it combines the flexibility of a neural network with the known dynamics captured by the simple parametric model, thus, introducing an inductive bias in the architecture.

Recent research has aimed to leverage the structure of dynamics and learn multiple models to complement each other. Huang and Tan [36], for instance, trained two neural networks specifically for modeling friction. Guo et al. [37] took this a step further by learning additional, individual neural networks for the inertia matrix, the Coriolis torque, and the gravitational torque, which were subsequently combined into a single neural network. They also adapted discontinuous activation functions for friction torque. Most of the work, however, does not capture the hysteresis effect of friction, as they lack a dependence on the history. To model this, Hirose and Tajima [38] applied a Long Short-Term Memory (LSTM) network, a type of recurrent neural network (RNN) that can naturally process sequences by maintaining a hidden state. Another approach was taken by Tu et al. [39], who used genetic algorithms (GA) to compute suitable initializations for the neural network weights. For human–robot interaction, it is crucial to be able to estimate the external torque of robot joints, which was the goal of Liu et al. [40] by approximating the friction using neural networks.

### 1.2. Our contributions

The application of data-driven approaches to model friction in robotic joints is not a novel concept. However, previous approaches have exhibited limitations in their generality due to the absence of important components within the data. Specifically, these approaches lack the incorporation of different velocities, the reversal of directions, simultaneous movement of joints, and continuously varying loads. These conditions significantly influence the performance of data-driven approaches, as evidenced in Section 4.1. The results demonstrate that a network, that may outperform traditional model-based approaches on a dataset lacking some of these effects, ultimately fails when subjected to more complex data. Furthermore, dynamics also change due to wear, varying temperature and humidity, and other external factors.

To address these challenges, we propose a novel strategy to adopt existing methods to unknown dynamics while requiring as little data as

possible—only one movement in our case. To achieve this we train a neural network on the residual of a base estimator on new dynamics for which the base estimator fails, to be able to use the knowledge learned by the base estimator while improving its performance for the new dynamics. The key contributions of this paper are as follows:

- 1. Adaption of a base model to new friction dynamics:** To tackle the challenge of adapting existing friction models to new dynamics with as little data as possible, we propose to learn the residuals using a neural network.
- 2. Comprehensive evaluation:** We evaluate our approach using a data-driven base model of friction and compare it against a conventional model-based approach, an extended LuGre model. We show that our approach then only requires data from a single point-to-point movement that includes a velocity reversal to be able to adapt the prediction of the friction torque and outperform the conventional approach and the base network for different velocities, while reversing the directions, moving the joints simultaneously, and continuously varying the external loads.
- 3. Integration into torque estimation framework:** To demonstrate the practical applicability of our adapted friction model, we integrate it into a torque estimation framework. By estimating the external torque applied to an object more precisely than the base network and the traditional approach, our approach enhances the overall precision and efficiency of robotic systems. We validate the accuracy of our torque estimation using external torque sensors, thereby ensuring the reliability of our proposed methodology.

The successful adaption of existing friction models and their integration into the torque estimation framework holds promise for numerous applications in the field of robotics. This approach can potentially advance dynamic simulations, friction compensation techniques, and external torque estimation methodologies, enabling the development of more capable and adaptable robotic systems across various domains. By being able to adapt existing models to new dynamics, this approach leverages the wealth of existing knowledge and models, making it a powerful tool for addressing complex and dynamic scenarios in robotics.

## 2. Methods

The goal of this paper is to introduce a novel approach to adapt existing methods to new dynamics using as little data as possible. This is desirable for several reasons, including time and cost savings, prioritizing data quality over quantity, resource limitations, and reduced complexity. In Section 2.1, we start with the introduction of the conventional model-based approach we use as a baseline for comparison throughout our experiments. Afterward, in Section 2.2, we describe the neural network-based approach we will use as our base model. In Section 2.3, we introduce our novel technique to adapt the neural network-based approach to unknown friction dynamics.

In the following, we describe the robot dynamics and our assumptions during the training of the different approaches to isolate the friction effect. Consider the robot dynamics of the form

$$M(q)\ddot{q} + C(q, \dot{q})\dot{q} + \tau_g(q) = \tau_m + \tau_f + \tau_{ext}, \quad (1)$$

where  $q$  is the joint position vector,  $\dot{q}$  and  $\ddot{q}$  are the joint velocity and acceleration vectors, respectively, and  $M(q)$  is the positive definite inertia matrix. The Coriolis and centrifugal matrix is denoted by  $C(q, \dot{q})$ , the gravitational torque by  $\tau_g(q)$ , and the friction torque by  $\tau_f$ . Furthermore, the terms  $\tau_m$  and  $\tau_{ext}$  describe the motor joint torque and external joint torque, respectively.

To predict the external torque using the estimation of the friction torque, we rewrite the dynamics equation as

$$\tau_{ext} = M(q)\ddot{q} + C(q, \dot{q})\dot{q} + \tau_g(q) - \tau_f - \tau_m. \quad (2)$$

We assume full knowledge of the robot dynamic terms ( $M(q)$ ,  $C(q, \dot{q})$ ,  $\tau_g(q)$ ), whereas  $\tau_f$  is unknown. The motor torque in this case is directly measured as  $\tau_m$ . If no load is applied ( $\tau_{ext} = 0$ ), the motor torque follows from the robot dynamics (assumed to be known) and the friction torque as

$$\tau_m = M(q)\ddot{q} + C(q, \dot{q})\dot{q} + \tau_g(q) - \tau_f. \quad (3)$$

In the special case where we assume constant, single-joint velocities or constant, low velocities ( $\ddot{q} = 0$ ,  $C(q, \dot{q})\dot{q} \approx 0$ ), the quadratic terms of individual joints can be ignored, while it is assured that the coupling terms are zero. As a result, the equation simplifies to

$$\tau_m = \tau_g(q) - \tau_f. \quad (4)$$

All the assumptions above will only be made during training to relate the motor torque to the friction effect as shown in Eq. (4).

### 2.1. Model-based approach

The friction torque  $\tau_f$  can be mathematically described by incorporating several fundamental characteristics that define friction in the sliding regime, including static friction, Coulomb friction, viscous friction, and the Stribeck effect [41]. In the pure sliding regime, the static friction behavior  $\tau_{f,s}$  can be represented by arbitrary functions, but a common model takes the form:

$$\begin{aligned} \tau_{f,s}(\dot{q}) &= g(\dot{q}) + s(\dot{q}) \\ g(\dot{q}) &= \text{sign}(\dot{q}) \left( F_c + (F_s - F_c) e^{-|\dot{q}/v_s|^{\delta_s}} \right). \end{aligned} \quad (5)$$

The term  $s(\dot{q})$  expresses the velocity-strengthening function that is well known as viscous friction. Typically, it is linearly proportional to the joint velocity  $\dot{q}$  as  $s(\dot{q}) = F_v \dot{q}$ , with the constant coefficient  $F_v$ . The velocity-strengthening friction effect could in the general case include a nonlinear form as shown in [42]. The function  $g(\dot{q})$  describes the velocity-weakening behavior of the static friction. Also,  $g(\dot{q})$  is alternatively called the Stribeck curve, because it captures the Stribeck effect, where  $F_c$  is Coulomb friction,  $F_s$  is static or stiction friction,  $v_s$  is Stribeck velocity, and  $\delta_s$  is the exponent parameter of the Stribeck-nonlinearity. The friction of the Harmonic-Drive (HD) gear-based robotic joint is known to be highly dependent on the temperature, which can be incorporated in the static and dynamic friction models [42,43]. Generally, the joint torque varies during the robot operation as it is configuration-dependent. This variation is reflected as a load effect in the joint friction torque, which can be included in Eq. (5) and results in

$$\tau_{f,s}(\dot{q}, \tau_1) = g(\dot{q}, \tau_1) + s(\dot{q}, \tau_1), \quad (6)$$

where  $\tau_{f,s}(\dot{q}, \tau_1)$  denotes the static friction torque as a function of velocity  $\dot{q}$  and load  $\tau_1$ , which is mainly gravitational torque in our case, for more details refer to [42]. As the friction phenomenon by nature is nonlinear and continuous at zero velocity crossing, it is unpractical to use the static friction model which is discontinuous at velocity reversal. Therefore, the dynamic friction effect can be expressed as

$$\tau_{f,d}(\dot{q}, \tau_1) = f(z, \dot{q}, \tau_1), \quad (7)$$

where  $\tau_{f,d}$  is the dynamic friction torque and  $z$  is the internal friction state with its dynamics

$$\frac{dz}{dt} = \dot{q} - \sigma_0 \frac{|\dot{q}|}{g(\dot{q}, \tau_1)} z. \quad (8)$$

Eq. (8) can be rewritten as

$$\tau_{f,d} = \sigma_0 z + \sigma_1 \dot{z} + s(\dot{q}, \tau_1). \quad (9)$$

The pre-sliding parameters are the bristle stiffness  $\sigma_0$  and the micro-damping coefficient  $\sigma_1$ , for more details see [44,45].

Eq. (9) represents a smooth and continuous expression that can describe the friction dynamically and extends the LuGre model [44] to

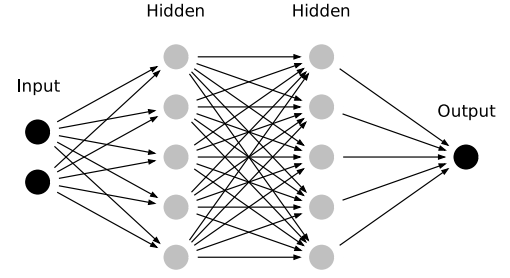


Fig. 1. The architecture of the base neural network: Fully connected neural network with 2 hidden layers, each consisting of 30 neurons (for space reasons, we only show five neurons here).

incorporate load dependency, which is considered in this work as the conventional model-based approach. While the bristles-based dynamic friction models show high accuracy in capturing the physical friction effects of the robotic joints, the difficulty of estimating and adapting their parameters remains. This limits the usage of such models in many scenarios and proves a strong motivation for data-driven approaches.

### 2.2. Neural network based approach

As our base model, we propose a data-driven model utilizing a neural network to learn the friction torque as a function of the gravity torque and its velocity. In our methodology, the focus lies on learning an unknown target function  $f : X \rightarrow Y$  through the observation of input-output pairs  $\{(x_i, y_i)\}_{i=1}^n \subset X \times Y$  with  $f(x_i) \approx y_i$ , where  $X$  denotes the input space and  $Y$  represents the output space. It is assumed, with some simplification, that the data points  $x_i$  are sampled from a sequence of independently and identically distributed (iid) random variables, governed by a common probability density function  $\mu$ , which is supported on  $X$ , i.e.  $x_i \sim \mu$  for all  $i = 1, \dots, n$  and  $\mu(X) = 1$ . Furthermore, it is acknowledged that the observed output  $y_i$  may potentially be corrupted by noise, thus introducing a degree of uncertainty.

Our task now is to find a parameterized mapping  $f_\theta : X \times \Theta \rightarrow Y$  (in our case, a neural network) that approximates the desired function  $f$ , where  $\Theta$  represents a finite-dimensional parameter space. This is achieved by identifying a suitable cost function  $C : X \times X \rightarrow \mathbb{R}$ , which quantifies the discrepancy between the predicted outputs of  $f_\theta$  and the true outputs, and minimizing the generalization error or risk, i.e.,

$$\mathbb{E}_{X \sim \mu}[C(f, f_\theta)]$$

over all  $\theta \in \Theta$ . The generalization error  $\mathbb{E}$  reflects the expected value of this cost function over the entire input space  $X$ , capturing the network's ability to generalize well to unseen data. By minimizing the generalization error, we strive to find the optimal set of parameters  $\theta \in \Theta$  that minimizes the discrepancy between  $f_\theta$  and the true function  $f$ , with the hope that  $f \approx f_\theta$ .

However, since, in general, we do not have access to the underlying probability measure  $\mu$  and solely have a finite number of observations of input-output pairs  $\{(x_i, y_i)\}_{i=1}^n \subset X \times Y$ , we instead minimize the empirical risk or training error given by

$$L(\theta) = \sum_{i=1}^n C(y_i, f_\theta(x_i))$$

over all  $\theta \in \Theta$ . An introduction to statistical learning theory can be found in [46].

In the framework of learning the torque friction, we choose a neural network as our parametric map denoted by  $NN(\cdot, \theta)$ , and the cost function is the squared difference of its inputs, i.e., the loss  $L$  is the mean squared error. A neural network is a function

$$NN = f_1 \circ f_2 \circ \dots \circ f_l$$

given by a composition of functions  $f_i$  (the layers)

$$f_i(x; \theta_i) = \sigma(W_i x + b_i) \quad (10)$$

where  $\theta_i = (W_i, b_i)$  are the parameters,  $\sigma : \mathbb{R} \rightarrow \mathbb{R}$  is the so-called activation function, and  $l \in \mathbb{N}$  is the number of layers of the neural network. Fig. 1 shows the architecture of a two-hidden layer neural network which will be used for learning the torque friction.

The idea in this paper is to train a neural network to model the friction torque by following (4). For this, we set  $f_\theta = NN_{base}(\cdot; \theta) : X \rightarrow Y$  with  $X = \mathbb{R}^2$  and  $Y = \mathbb{R}$ , thereby allowing the neural network to depend on the gravitational torque  $\tau_g(q(t_i))$  and the velocity  $\dot{q}(t_i)$  at a given time step  $t_i$ .

As mentioned before, we then train the neural network by minimizing the mean squared error, i.e., the loss function is given by

$$L_{base}(\theta) = \sum_{i=0}^n (NN_{base}(\tau_g(q(t_i)), \dot{q}(t_i); \theta) - (\tau_m(t_i) - \tau_g(q(t_i))))^2. \quad (11)$$

$NN_{base}$  is a neural network with 2 hidden layers, each consisting of 30 neurons, as shown in Fig. 1, which we will refer to as the base model in the following. As the activation function, we use the following exponential linear unit (ELU) function, given by,

$$f(x) = \begin{cases} x, & x > 0 \\ e^x - 1, & x \leq 0 \end{cases} \quad (12)$$

which is a smoothed version of the rectified linear unit (ReLU) function.

### 2.3. Adaption to new dynamics

Since most models of friction are best for the dynamics they are built and trained for, it is helpful if existing models can be easily extended to new situations without the need for big data sets to retrain the models completely. For that reason, we propose to build upon the existing models to not waste the previously acquired knowledge and train an additional neural network  $NN_{add}$  on the residuals of the old model on the new data. Therefore, we minimize the following loss:

$$L_{add}(\phi) = \sum_{i=0}^n \left( NN_{add}(\tau_g^{add}(q(t_i)), \text{sign}(\dot{q}^{add}(t_i)); \phi) + \tau_{f_{base}}(\tau_g^{add}(q(t_i)), \dot{q}^{add}(t_i)) - (\tau_m^{add}(t_i) - \tau_g^{add}(q(t_i))))^2. \quad (13)$$

We allow the network  $NN_{add}$  to depend on the gravity torque and the sign of the velocity;  $\phi$  denotes the parameters of  $NN_{add}$  we want to optimize and  $\tau_{f_{base}}$  is the base model. In general, any base model works for  $\tau_{f_{base}}$ , however, in this study, we utilize  $NN_{base}$  as our base model, i.e., we set  $\tau_{f_{base}} = NN_{base}$ .

It is important to note that we only use the sign of the velocity as input for  $NN_{add}$ , rather than the full velocity. This choice is motivated by our objective to minimize the amount of data required for training. In our case, we will only use data from one movement with one velocity in both directions, which is insufficient to fully capture the complete velocity dependence. Nevertheless, this is not a problem because the base model,  $NN_{base}$ , already accounts for the velocity dependence. Note further that using the sign of the velocity fits to the discontinuity of the friction at zero velocity.

The new predictor for the small friction data set with varying load and directions is then

$$\tau_{f_{pred}}(q, \dot{q}) = \tau_{f_{base}}(q, \dot{q}; \theta^*) + NN_{add}(q, \text{sign}(\dot{q}); \phi^*), \quad (14)$$

where  $\phi^*$  denotes the parameters of the additional model, found when minimizing  $L_{add}(\phi)$ .

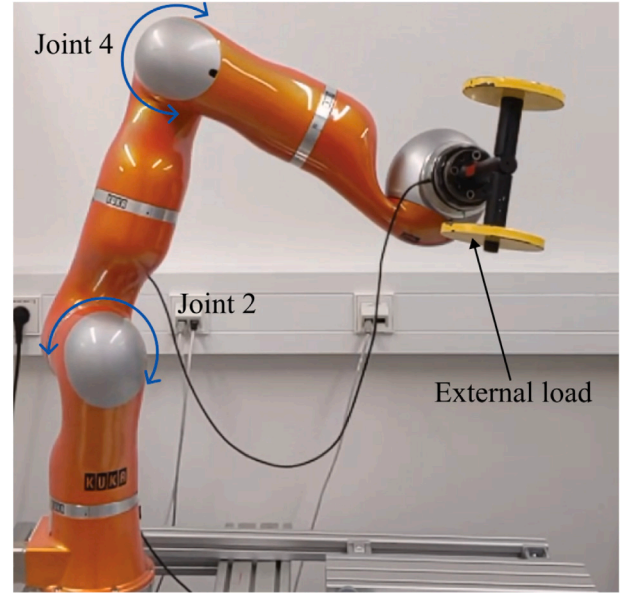


Fig. 2. The experimental setup: the robot is equipped with physical torque sensors in each joint output for the reference signals, while the measured motor current is used in the proposed method.

### 3. Data sets

Since the goal of this work is to adapt an existing approach to a set with different dynamics, we utilize two different data sets. Both data sets are collected to best capture the physical behavior of the friction torque. The experimental measurements are acquired from each robot joint separately at different constant velocities. In this study, the torque-controlled DLR-KUKA LWR-IV+ robot is used as a reference platform. The full experimental setup is illustrated in Fig. 2.

This work aims to model the friction effects mainly resulting from the gear (HD), however, the used robot is also equipped with link-side torque sensors, which are used for validation only. The data is collected in two main subsets to best describe the friction effect and eliminate other unmodeled dynamics.

For each data set we collect the position and velocity of the joints and the gravitational torque and applied motor current. The motor current  $I_m$  is used to compute the motor torque  $\tau_m = K_t I_m$ , where  $K_t$  is the motor torque constant taken from the manufacturer's data sheet. The gravitational torques in the robot joints are obtained through a model-based approach using recursive Newton-Euler formulation which relies on the robot's mechanical design information, e.g., the mass and center of mass of each link, for which we rely on an accurate CAD model.

Notably, the load variation effect also appears in the no-load data as a result of gravitational torque in the robot joints. This can be seen from the measured data in Figs. 6(b) and 6(c) in Section 3.2 as the friction torque varies even when the velocities are constant. The temperature effect is minimized by applying warm-up phases before each data collection routine. In this way, the temperature is assumed to be constant during operation.

#### 3.1. Base data set

The base data set aims to capture the static friction behavior; therefore, the robot joints were excited to follow single-joint constant velocities. Thus, (4) holds and can be used to compute the friction. We will refer to this data set as the collected friction data set. Intensive experimental measurements have been carried out to cover the entire operational velocity and position ranges of the individual joints; the

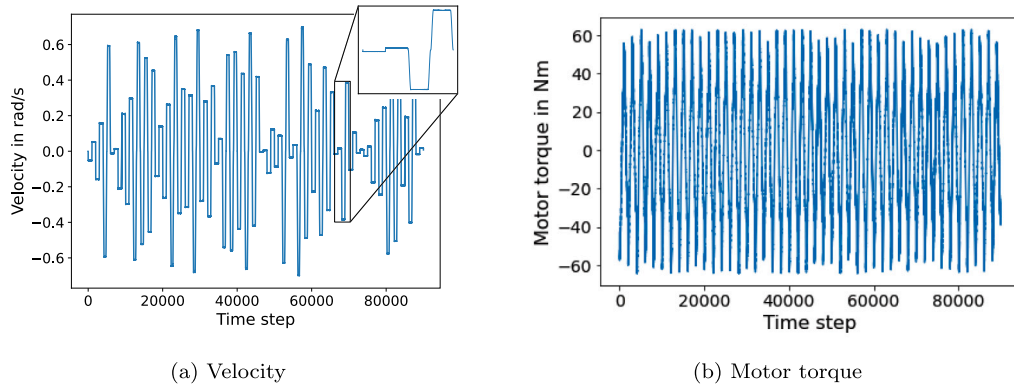


Fig. 3. Velocity and motor torque of the collected friction dataset for Joint 2.

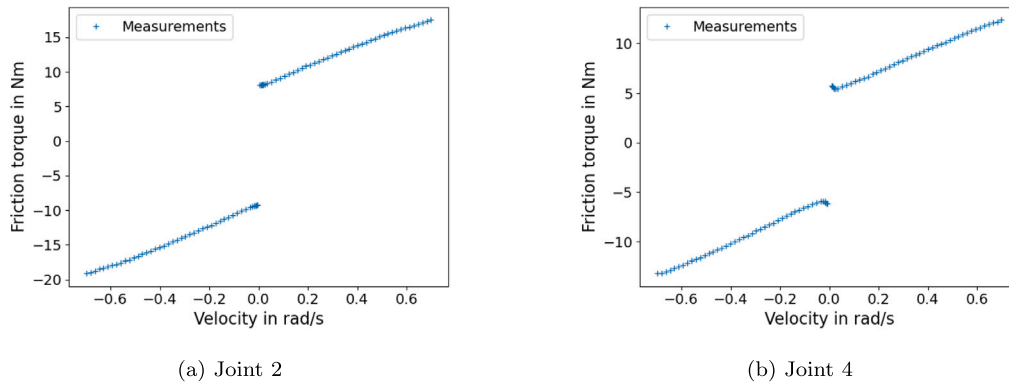


Fig. 4. The measured friction torque-velocity behavior for Joint 2 and Joint 4 of the DLR-KUKA LWR-IV+ robot for the base data set.

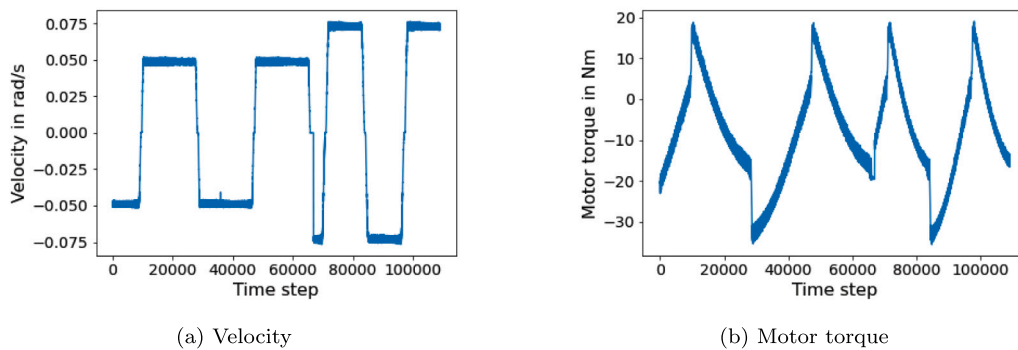


Fig. 5. Velocity and motor torque of the small dataset with varying directions and without external load for Joint 2.

velocity and motor torque of Joint 2 can be seen in Fig. 3. In this way, the static friction is characterized in a fine resolution and separated from other effects as shown in Fig. 4.

### 3.2. Extended data set

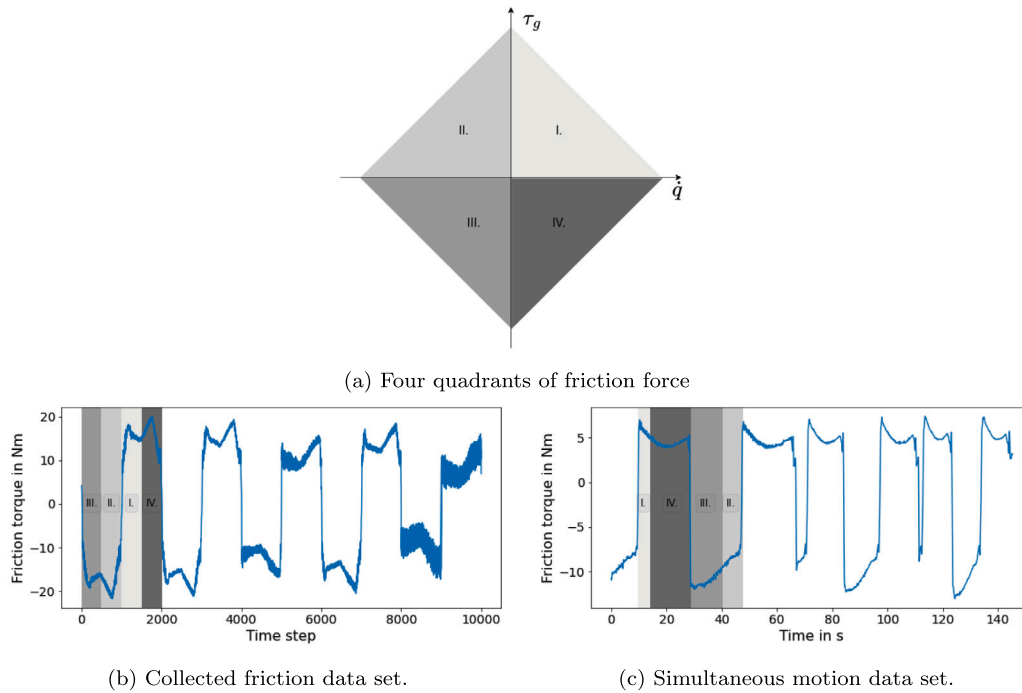
The extended data set is collected dynamically by applying different velocities sequentially while different external loads are attached to the robot end-effector and the joints are moved simultaneously. We refer to this data set as the small data set with varying loads and directions. The velocity and motor torque for this dataset without external torque are displayed in Fig. 5. By keeping the velocity low and constant we ensure that (4) can be used to compute the friction torque.

The behavior of friction torque in robotic joints varies across the four quadrants, as defined in Fig. 6(a), due to changes in the interaction between velocity and gravitational torque [47]: In the first and third

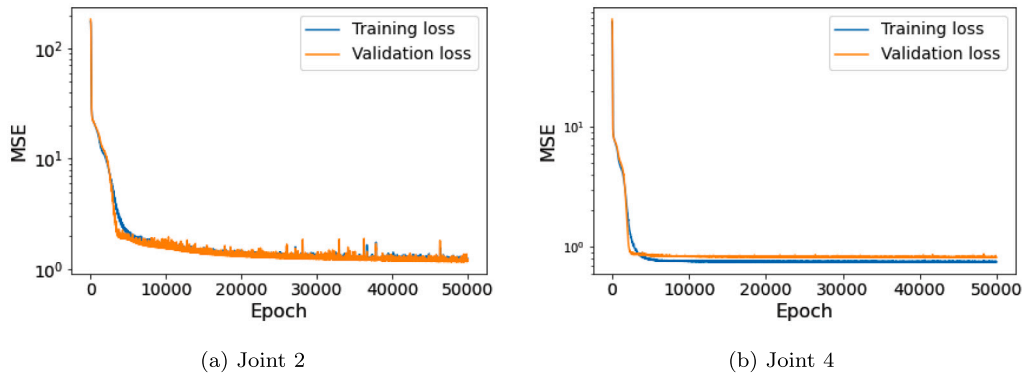
quadrants, they have the same sign, while they have different signs in the second and fourth. The difference between the two datasets investigated in this paper can be visualized by examining the quadrants during one movement from each dataset, as shown in Fig. 6. It is evident that the base dataset features entirely symmetric quadrants, whereas the additional dataset demonstrates a highly asymmetric trajectory across the quadrants.

## 4. Experimental results

In this section, we present the results of the numerical experiments. In Section 4.1, we show the results of  $NN_{base}$  on both the symmetric and asymmetric datasets and compare the performance to the conventional approach. These experiments reveal that the neural network-based approach performs similarly to the conventional approach and both struggle with the asymmetric data set. Consequently,



**Fig. 6.** Four quadrants of friction force: The different grey tones show the different quadrants the friction is going through. The friction stays in each quadrant for the exact same time in the collected friction data set, but the times vary strongly for the simultaneous motion data set.



**Fig. 7.** The training and validation loss  $L_{base}$  for Joint 2 and Joint 4 for  $NN_{base}$ .

In Section 4.2, we demonstrate the improvements achieved by adapting  $NN_{base}$  using  $NN_{add}$ .

#### 4.1. Prediction of the base model

The base network  $NN_{base}$  is trained on a subset of the collected friction data set using the Adam optimizer [48] with a learning rate of 0.01 for 50,000 training steps. Since for each velocity, the joint moves exactly once in both directions, the data contains more measurements in the low-velocity regime than in the high-velocity regime. As a consequence, the loss function tends to prioritize the low-velocity region, resulting in high precision for low velocities and low precision for high velocities. To address this, we downsample the data, ensuring an equal number of data points for each velocity, as depicted in Fig. 3. Besides balancing the precision across velocity ranges, this downsampling also significantly improves runtime, particularly in the low-velocity regime where many data points are very similar.

Fig. 7 displays the behavior of the loss function  $L_{base}$  on both the training and validation data during the training of  $NN_{base}$  for Joint

2 and 4, illustrating that the model effectively converges robustly to a local minimum and can be expected to generalize well, since the validation loss is close to the training loss. To assess the performance of the base neural network  $NN_{base}$  trained on the collected friction data set, Fig. 8 presents the comparison between its prediction and the measurements. The network demonstrates its capability to accurately approximate the friction torque, showcasing its effectiveness. In the next step, we explore whether the data-driven model  $NN_{base}$  learned a meaningful relationship between velocity and friction. To investigate this, we evaluate the model on a grid spanning from  $-0.7$  to  $0.7$  rad/s for the velocity and from  $-43$  to  $43$  N m for the gravitational torque for Joint 2 ( $-13$  to  $13$  N m for Joint 4). Fig. 9 visualizes this evaluation as both a heatmap and a 3D plot, while Fig. 10 presents the averaged results over the gravitational torque. These plots clearly demonstrate that the base neural network has learned a physically meaningful friction model, suggesting strong potential for generalization, as further supported by the results in Fig. 8.

As the neural network effectively captured the dynamics of friction for the base data set, the next step is to assess its performance on the

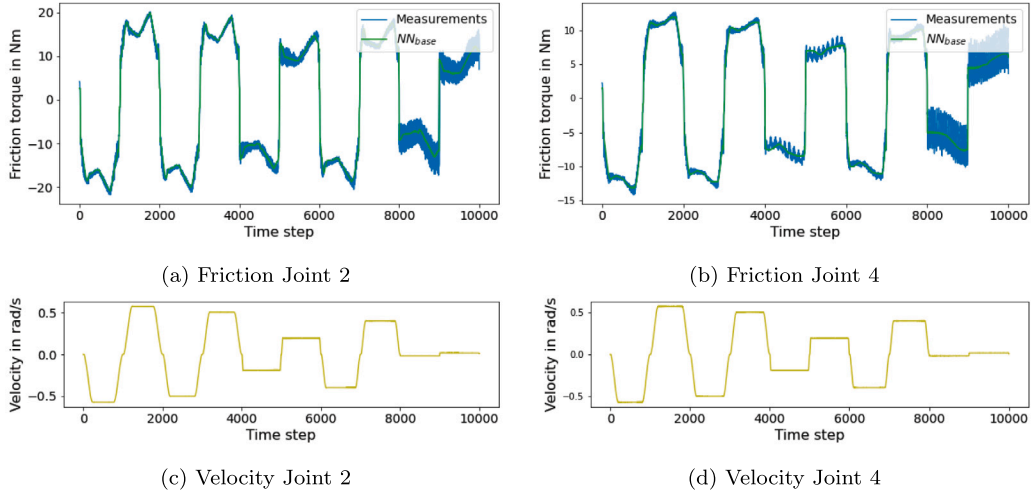


Fig. 8. Figures (a) and (b) show the true friction torque of the collected friction data for Joints 2 and 4, respectively, which was withheld during the training and validation phase, and the estimate of the neural network. The mean absolute error was 0.79 N m and 0.60 N m, respectively. Figures (c) and (d) show the velocity for Joints 2 and 4, respectively.

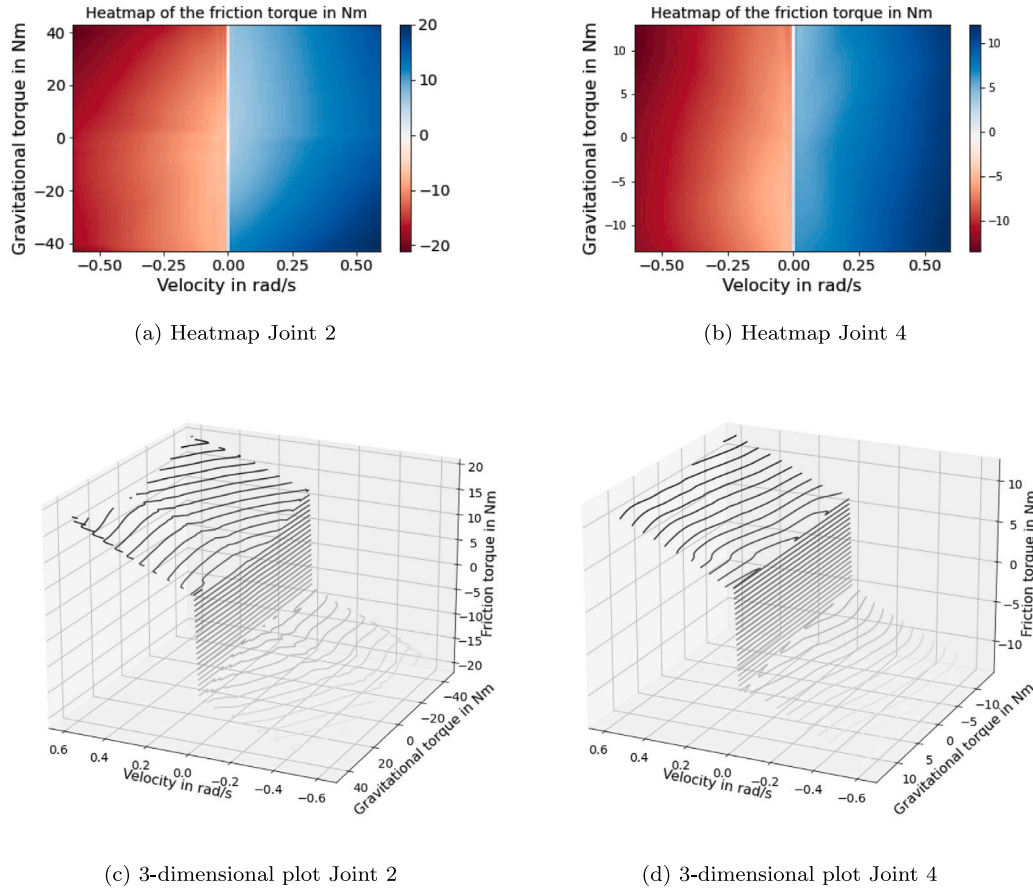


Fig. 9. Dependence of the friction on the velocity and the position as captured by the base model  $NN_{base}$  for Joint 2 and 4.

extended data set. Figs. 11 and 12 reveal that the base network  $NN_{base}$  outperforms the conventional method, but it performs significantly worse than it did for the collected friction data set. In Fig. 11(e) and (f), as well as Fig. 12(e) and (f), the friction estimates are utilized to predict the external torque, illustrating the prediction error introduced through the error in friction modeling.

#### 4.2. Prediction of the adapted model

To improve the prediction of  $NN_{base}$  on the extended data set, we train the additive network  $NN_{add}$  as described in Section 2.3 by minimizing  $L_{add}$  in Eq. (13). Specifically, we provide only one velocity from the small data set without external load as training data. This does not lead to overfitting, since  $NN_{add}$  depends solely on the gravitational

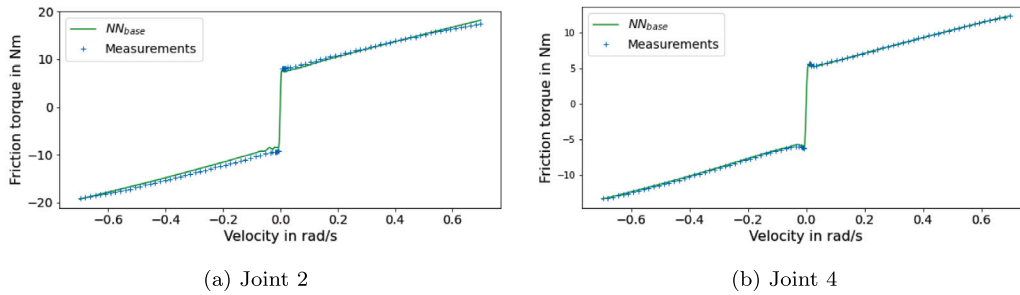


Fig. 10. Dependence of the friction on the velocity as modeled by the base model  $NN_{base}$ .

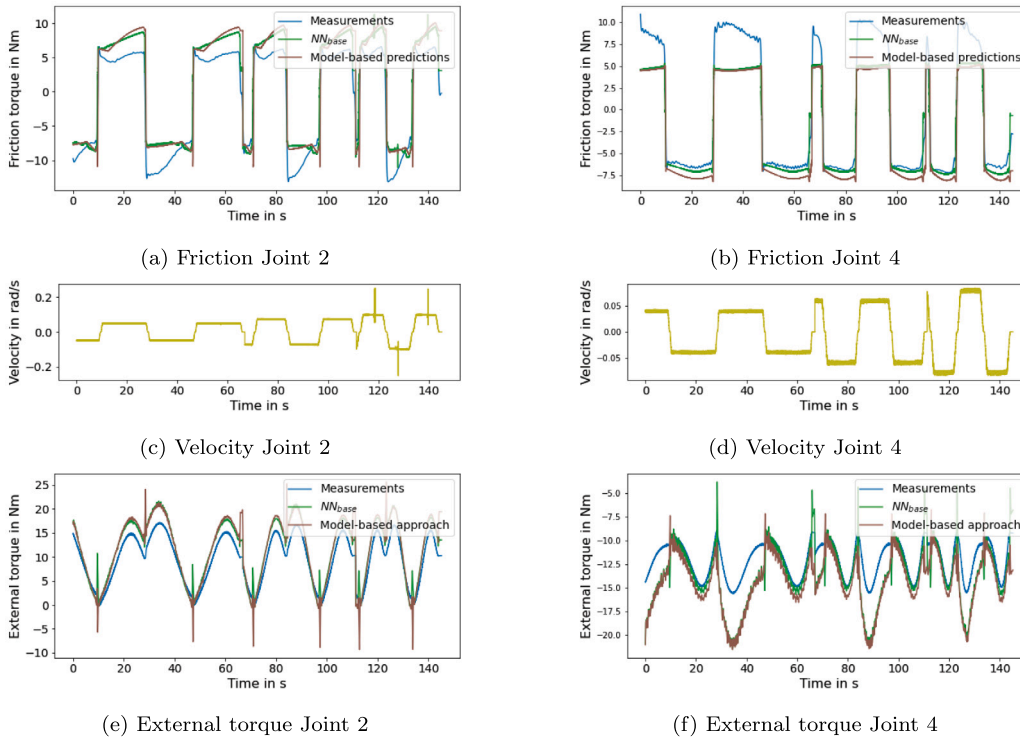


Fig. 11. Performance of the neural network versus the model-based approach on the small data set with asymmetric load. In (a) and (b) the friction is estimated for Joint 2 and 4. On Joint 2, the neural network achieved an average error of 2.40 N m, while the conventional approach had an average error of 2.68 N m. On Joint 4, the neural network achieved an average error of 1.91 N m, while the conventional approach had an average error of 2.45 N m. (c) and (d) show the velocities of the respective joints. In (e) and (f) the friction estimates are used to estimate the external torque, which is shown after denoising.

torque and the sign of the velocity. The benefit of using only data without external load during training is that this does not require any external torque sensor. The architecture of the additional model  $NN_{add}$  mirrors that of the base model, however, it only uses one hidden layer. The training process is also the same, but  $NN_{add}$  is trained for only 200 epochs. Fig. 13 displays the training and validation loss during training, showcasing robust minimization across epochs, but a slight difference in training and validation loss.

Fig. 14 demonstrates the accuracy of the friction modeling achieved by the adapted approach, as described in Section 2.3, in comparison to  $NN_{base}$  and the conventional approach, even though only one velocity was observed during the additional training period (43s). Furthermore, Figs. 15 and 16 reveal that this extension of the original neural network has a minimal impact on the velocity dependence, particularly preserving the underlying relationship that would be lost through the retraining of any network components.

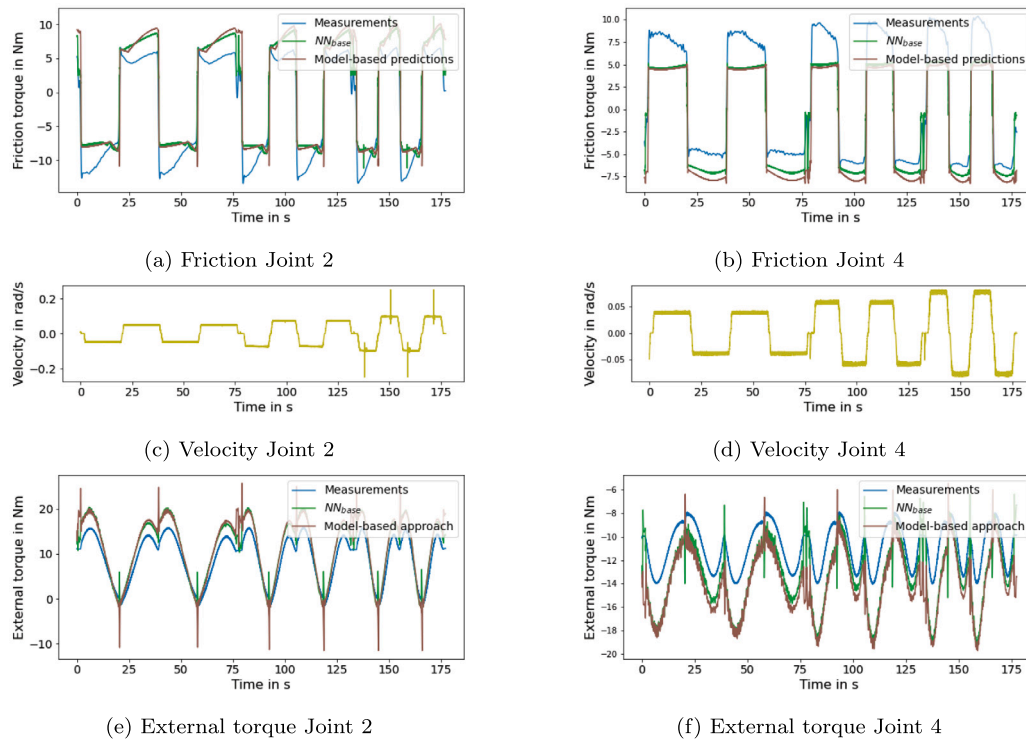
The combined approach not only exhibits a significant improvement for the small data set with varying directions when no external load is applied but also when asymmetric and symmetric external loads

are present, as demonstrated in Figs. 17(a) and (b) and 18(a) and (b), respectively. By leveraging the accurate prediction of the adapted method for the friction torque we can estimate the external torque via the robot dynamics. Contrary to the prediction using only  $NN_{base}$  in Figs. 11(e) and (f) and 12(e) and (f), this estimate strongly outperforms the conventional approach, see Figs. 17(e) and (f) and 18(e) and (f) for the asymmetric and symmetric load cases, respectively.

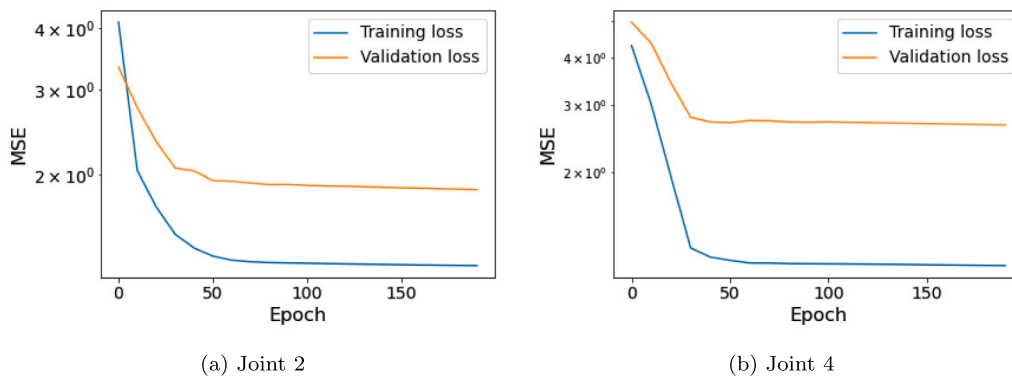
#### 4.3. Comparisons with other data-driven methods

To the best of our knowledge, this is the first work focusing on the adaptation of existing friction estimation models. Therefore, we primarily compared the base method with the adapted method, which is method-agnostic and can be applied to other methods presented in the literature. To ensure clarity, we concentrated in Sections 4.1 and 4.2 on classical methods (vanilla neural network and LuGre model), highlighting the benefits of residual learning.

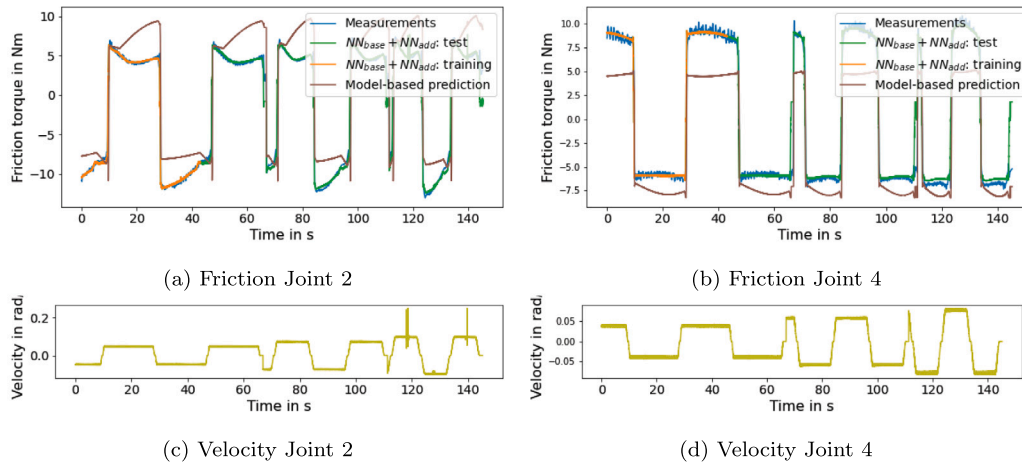




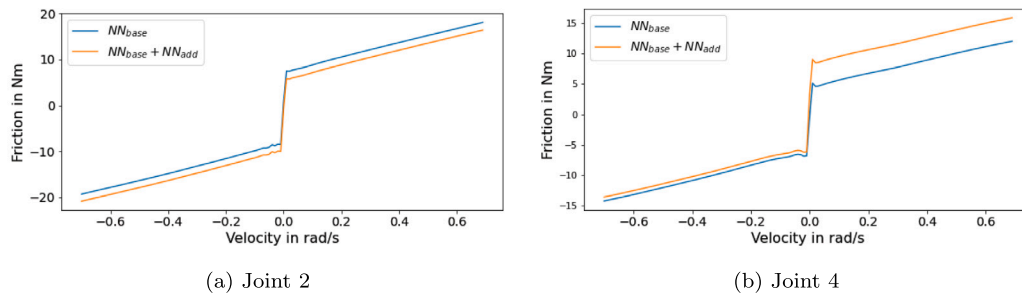
**Fig. 12.** Performance of the base model  $NN_{base}$  versus the model-based approach on the small data set with symmetric load. In (a) and (b) the friction is estimated for Joint 2 and 4. On Joint 2, the base model achieved an average error of 2.51 N m, while the conventional approach had an average error of 2.89 N m. On Joint 4 the base model achieved an average error of 2.48 N m, while the conventional approach had an average error of 3.1 N m. (c) and (d) show the velocities of the respective joints. In (e) and (f) the friction estimates are used to estimate the external torque, which is shown after denoising.



**Fig. 13.** The training and validation loss  $L_{add}$  for Joints 2 and 4 for  $NN_{add}$ .



**Fig. 14.** Performance of the combined network  $NN_{base} + NN_{add}$  versus the model-based approach on the small data set without external load. In (a) and (b) the friction is estimated for Joint 2 and 4. On Joint 2, the combined neural network achieved an average error (on the test data) of 0.48 N m, while the conventional approach had an average error of 2.62 N m. On Joint 4, the neural network achieved an average error of 0.66 N m, while the conventional approach had an average error of 2.33 N m. (c) and (d) show the velocities of the respective joints.



**Fig. 15.** Dependence of the friction on the velocity as modeled by the combined neural network  $NN_{base} + NN_{add}$  versus as modeled by the base neural network.

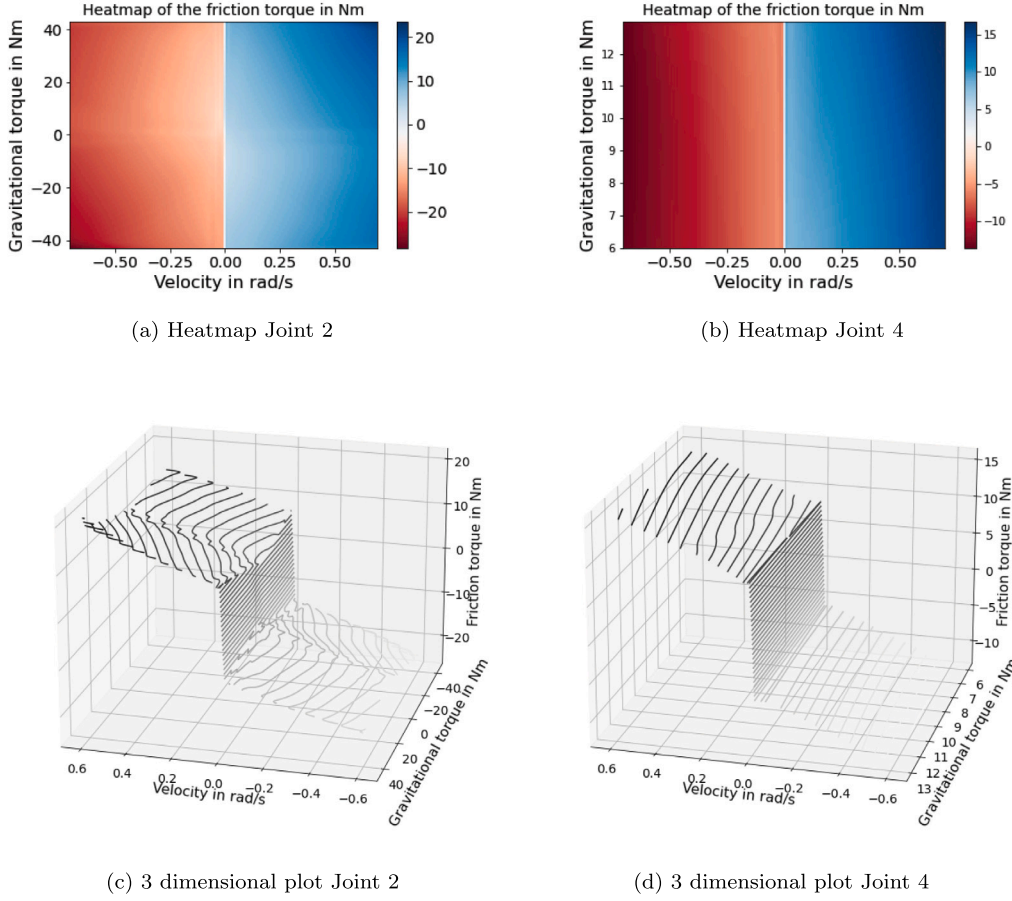


Fig. 16. Dependence of the friction on the velocity and the position as captured by the combined neural network  $NN_{base} + NN_{add}$  for Joint 2 and 4.

However, a comparative analysis with other data-driven approaches, as outlined in the literature review in Section 1.1, is of significant interest. To accomplish this, we re-implemented the methods proposed by Semic and Lewis [33] and by Ciliz and Tomizuka [34,35].

Semic and Lewis [33] introduced a fully connected neural network with discontinuous activation functions to model the discontinuity of friction at zero velocity. For this purpose they applied the standard Sigmoid functions  $\sigma(x) = \frac{1}{1+e^{-x}}$  and additionally the Sigmoid jump approximation functions

$$\phi_k(x) = \begin{cases} 0, & \text{for } x < 0, \\ (1 - e^{-x})^k, & \text{for } x \geq 0. \end{cases} \quad (15)$$

An extensive hyperparameter search, similar to the one performed for  $NN_{base}$ , showed that the network works best with 1 hidden layer comprising 30 neurons. Among these neurons, 20 used the standard Sigmoid as the activation function, while the remaining neurons applied the Sigmoid jump approximation function, with  $k$  varying from 1 to 10 for each neuron, as proposed by Semic and Lewis [33]. The network was optimized using ADAM with a learning rate of 0.01 over 50,000 steps.

Ciliz and Tomizuka [34,35] combine a neural network with a parametric approach to incorporate the flexibility of data-driven approaches with common knowledge about friction by modeling the friction torque  $\tau_f$  as

$$\hat{\tau}_f(\dot{q}, \tau_q) = NN(\dot{q}, \tau_q) + \beta_1 0.5(1 + \text{sign}(\dot{q})) + \beta_2 0.5(1 - \text{sign}(\dot{q})) \quad (16)$$

named *neural network with adaptive coulomb friction (NNACMF)*. Here,  $\beta_1$  and  $\beta_2$  denote learnable parameters. We extended the original NNACMF

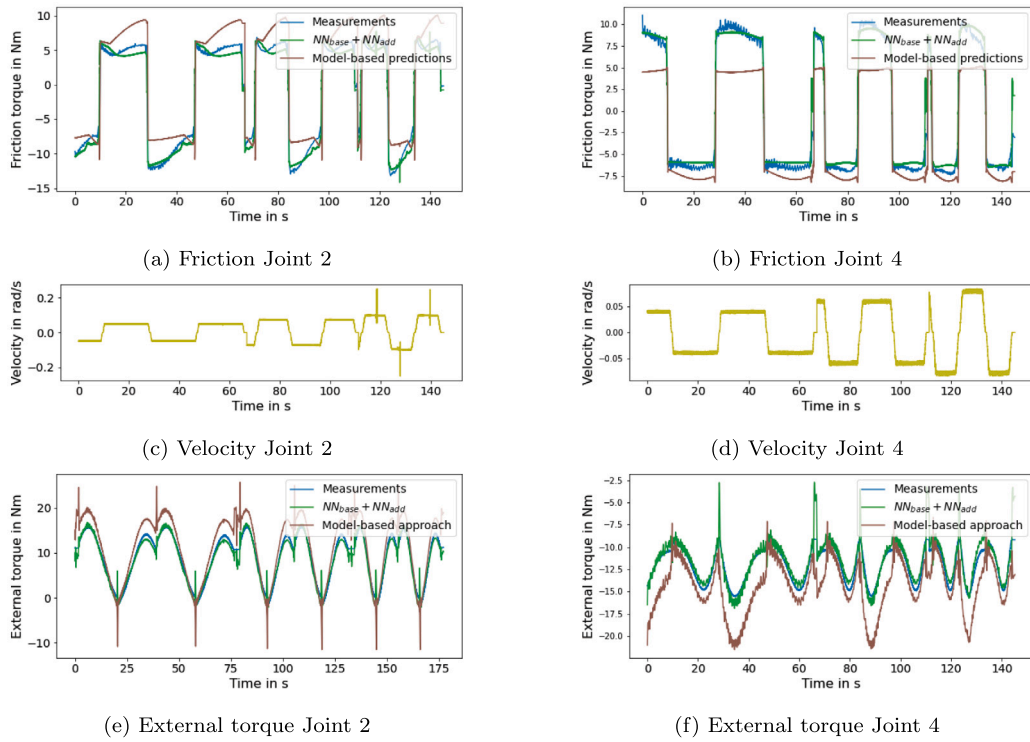
Table 2

Comparisons between the base methods from Sections 2.1 and 2.2 and the adapted approach from Section 2.3 with the approaches from Semic and Lewis [33] and Ciliz and Tomizuka [34,35] on joint 2.

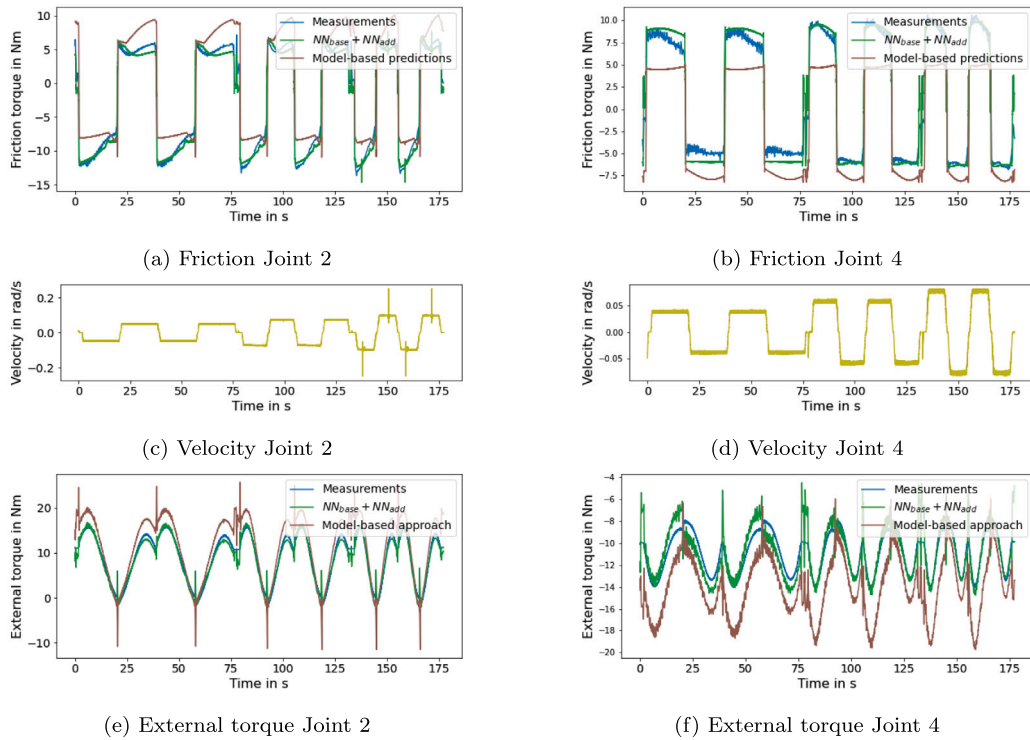
Method	Error with the asymmetric load	Error with the symmetric load
Conventional model (ext. LuGre)	2.68 N m	2.89 N m
$NN_{base}$	2.40 N m	2.51 N m
Semic and Lewis [33]	2.17 N m	2.37 N m
Ciliz and Tomizuka [34,35]	2.33 N m	2.42 N m
$NN_{base} + NN_{add}$	<b>0.87 N m</b>	<b>0.80 N m</b>

model to allow the neural network to depend on the gravitational torque, as shown in (16), to ensure a fair comparison with the other approaches presented in this work. Through an extensive hyperparameter search, we determined that using one hidden layer with 50 neurons yielded optimal results. The network was optimized using ADAM with a learning rate of 0.01 over 50,000 steps.

The results of both methods on the extended data set with asymmetric and symmetric loads can be observed in Table 2. It is evident from the table that both approaches perform slightly better than the extended LuGre model and the base neural network. However, the combined neural network proposed to learn the new dynamics still outperformed them by a margin, suggesting that these approaches also struggled with the new dynamics.



**Fig. 17.** Performance of the combined neural network  $NN_{base} + NN_{add}$  versus the model-based approach on the asymmetric load data. In (a) and (b) the friction is estimated for Joint 2 and 4. On Joint 2, the combined neural network achieved an average error of 0.87 N m, while the conventional approach had an average error of 2.66 N m. On Joint 4, the neural network achieved an average error of 0.70 N m, while the conventional approach had an average error of 2.45 N m. (c) and (d) show the velocities of the respective joints. In (e) and (f) the friction estimates are used to estimate the external torque, which is shown after denoising.



**Fig. 18.** Performance of the combined neural network  $NN_{base} + NN_{add}$  versus the model-based approach on the symmetric load data. In (a) and (b) the friction is estimated for Joint 2 and 4. On Joint 2, the combined neural network achieved an average error of 0.80 N m, while the conventional approach had an average error of 2.88 N m. On Joint 4, the neural network achieved an average error of 0.86 N m, while the conventional approach had an average error of 3.09 N m. (c) and (d) show the velocities of the respective joints. In (e) and (f) the friction estimates are used to estimate the external torque, which is shown after denoising.

## 5. Summary and conclusion

Due to the lack of a precise mathematical description of friction torque in robotic joints, model-based approaches struggle to capture its behavior accurately. This, in turn, hinders robot control in various situations, especially when dealing with new movements or environments. To address this challenge, we proposed an approach that adapts existing friction models to new dynamics using a minimal amount of data, which enhances efficiency, prioritizes data quality over quantity, and addresses resource limitations.

Our method leverages neural networks to learn the residuals of a base model on new dynamics, significantly improving model accuracy with only a short trajectory following the new dynamics. Importantly, our approach does not require specific domain knowledge and eliminates the need for external torque sensors, reducing the overall cost of robotic systems. Although this paper primarily demonstrates the use of a neural network as the base model, Eq. (13) permits the adaptation of any base model, thus enabling the incorporation of specialized data-driven or model-based techniques. Our method has been thoroughly tested with highly detailed data encompassing diverse velocities, loads, and movement directions, and its predictions have been validated using torque sensors.

However, neural networks' black-box nature and occasional difficulties in generalizing to out-of-distribution data remain challenges. Addressing these issues while retaining the advantages of data-driven friction torque learning will be important areas for future research. Besides this, we consider the incorporation of the adaption approach proposed in this work into an online learning procedure as a valuable future direction.

### CRedit authorship contribution statement

**Philipp Scholl:** Software, Validation, Visualization, Writing – original draft, Writing – review & editing, Conceptualization, Formal analysis, Investigation, Methodology. **Maged Iskandar:** Conceptualization, Data curation, Formal analysis, Investigation, Methodology, Validation, Visualization, Writing – original draft. **Sebastian Wolf:** Data curation, Investigation, Validation, Writing – review & editing. **Jinoh Lee:** Data curation, Investigation, Validation, Writing – review & editing. **Aras Bacho:** Conceptualization, Investigation, Methodology, Validation, Writing – original draft, Writing – review & editing. **Alexander Dietrich:** Validation, Writing – review & editing, Conceptualization, Data curation, Investigation, Methodology. **Alin Albu-Schäffer:** Data curation, Funding acquisition, Project administration, Resources, Supervision, Validation, Writing – review & editing. **Gitta Kutyniok:** Funding acquisition, Project administration, Resources, Supervision, Validation, Writing – review & editing.

### Declaration of competing interest

The authors declare the following financial interests/personal relationships which may be considered as potential competing interests: Philipp Scholl reports financial support was provided by Federal Ministry of Education and Research Berlin Office. Gitta Kutyniok reports was provided by Federal Ministry of Education and Research Berlin Office. Gitta Kutyniok reports was provided by German Research Foundation. Maged Iskander, Alexander Dietrich, Jinoh Lee reports financial support was provided by German Aerospace Center DLR Oberpfaffenhofen. If there are other authors, they declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

### Data availability

The data that has been used is confidential.

## Declaration of Generative AI and AI-assisted technologies in the writing process

During the preparation of this work the authors used ChatGPT in order to improve readability and language. After using this tool/service, the authors reviewed and edited the content as needed and take full responsibility for the content of the publication.

## Acknowledgments

P. Scholl and G. Kutyniok acknowledge support by LMUexcellent, funded by the Federal Ministry of Education and Research (BMBF) and the Free State of Bavaria under the Excellence Strategy of the Federal Government and the Länder as well as by the Hightech Agenda Bavaria. Furthermore, G. Kutyniok was supported in part by the DAAD programme Konrad Zuse Schools of Excellence in Artificial Intelligence, sponsored by the Federal Ministry of Education and Research. G. Kutyniok also acknowledges support from the Munich Center for Machine Learning (MCML) as well as the German Research Foundation under Grants DFG-SPP-2298, KU 1446/31-1 and KU 1446/32-1 and under Grant DFG-SFB/TR 109 and Project C09.

M. Iskandar, A. Dietrich, and J. Lee were supported by MOTIE/KEIT under Grants 20014485 and 37520014398.

## References

- [1] A. Ajoudani, A.M. Zanchettin, S. Ivaldi, A. Albu-Schäffer, K. Kosuge, O. Khatib, *Progress and prospects of the human–robot collaboration*, *Auton. Robots* 42 (2018) 957–975.
- [2] W. Kim, L. Peternel, M. Lorenzini, J. Babič, A. Ajoudani, *A human-robot collaboration framework for improving ergonomics during dexterous operation of power tools*, *Robot. Comput.-Integr. Manuf.* 68 (2021) 102084.
- [3] A. Angleraud, A. Ekrekli, K. Samarawickrama, G. Sharma, R. Pieters, *Sensor-based human–robot collaboration for industrial tasks*, *Robot. Comput.-Integr. Manuf.* 86 (2024) 102663, <http://dx.doi.org/10.1016/j.rcim.2023.102663>.
- [4] B. Liu, P. Rocco, A.M. Zanchettin, F. Zhao, G. Jiang, X. Mei, *A real-time hierarchical control method for safe human–robot coexistence*, *Robot. Comput.-Integr. Manuf.* 86 (2024) 102666, <http://dx.doi.org/10.1016/j.rcim.2023.102666>.
- [5] P. Aivaliotis, S. Aivaliotis, C. Gkourmelos, K. Kokkalis, G. Michalos, S. Makris, *Power and force limiting on industrial robots for human-robot collaboration*, *Robot. Comput.-Integr. Manuf.* 59 (2019) 346–360, <http://dx.doi.org/10.1016/j.rcim.2019.05.001>.
- [6] A. Cherubini, R. Passama, A. Crosnier, A. Lasnier, P. Fraise, *Collaborative manufacturing with physical human–robot interaction*, *Robot. Comput.-Integr. Manuf.* 40 (2016) 1–13, <http://dx.doi.org/10.1016/j.rcim.2015.12.007>.
- [7] M. Iskandar, C. Ott, A. Albu-Schäffer, B. Siciliano, A. Dietrich, *Hybrid force-impedance control for fast end-effector motions*, *IEEE Robot. Autom. Lett.* (2023).
- [8] M. Hägele, K. Nilsson, J.N. Pires, R. Bischoff, *Industrial robotics*, in: B. Siciliano, O. Khatib (Eds.), *Springer Handbook of Robotics*, Springer International Publishing, Cham, 2016, pp. 1385–1422, [http://dx.doi.org/10.1007/978-3-319-32552-1\\_54](http://dx.doi.org/10.1007/978-3-319-32552-1_54).
- [9] M.R. Kermani, R.V. Patel, M. Moallem, *Friction identification and compensation in robotic manipulators*, *IEEE Trans. Instrum. Meas.* 56 (6) (2007) 2346–2353.
- [10] A. Dietrich, X. Wu, K. Bussmann, M. Harder, M. Iskandar, J. Engelsberger, C. Ott, A. Albu-Schäffer, *Practical consequences of inertia shaping for interaction and tracking in robot control*, *Control Eng. Pract.* 114 (2021) 104875.
- [11] F. Xie, Z. Chong, X.-J. Liu, H. Zhao, J. Wang, *Precise and smooth contact force control for a hybrid mobile robot used in polishing*, *Robot. Comput.-Integr. Manuf.* 83 (2023) 102573, <http://dx.doi.org/10.1016/j.rcim.2023.102573>.
- [12] X. Ke, Y. Yu, K. Li, T. Wang, B. Zhong, Z. Wang, L. Kong, J. Guo, L. Huang, M. Idir, C. Liu, C. Wang, *Review on robot-assisted polishing: Status and future trends*, *Robot. Comput.-Integr. Manuf.* 80 (2023) 102482, <http://dx.doi.org/10.1016/j.rcim.2022.102482>.
- [13] D.K.-G. P. Aivaliotis, S. Makris, *Physics-based modelling of robot's gearbox including non-linear phenomena*, *Int. J. Comput. Integr. Manuf.* 36 (12) (2023) 1864–1875, <http://dx.doi.org/10.1080/0951192X.2022.2162594>, arXiv:<https://doi.org/10.1080/0951192X.2022.2162594>.
- [14] J. Reinecke, A. Dietrich, F. Schmidt, M. Chalon, *Experimental comparison of slip detection strategies by tactile sensing with the BioTac<sup>®</sup> on the DLR hand arm system*, in: *2014 IEEE International Conference on Robotics and Automation, ICRA, IEEE, 2014*, pp. 2742–2748.
- [15] M. Iskandar, C. Ott, O. Eiberger, M. Keppler, A. Albu-Schäffer, A. Dietrich, *Joint-level control of the DLR lightweight robot SARA*, in: *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS, IEEE, pp. 8903–8910*.

- [16] M. Iskandar, C. van Ommeren, X. Wu, A. Albu-Schäffer, A. Dietrich, Model predictive control applied to different time-scale dynamics of flexible joint robots, *IEEE Robot. Autom. Lett.* 8 (2) (2022) 672–679.
- [17] A. De Santis, B. Siciliano, A. De Luca, A. Bicchi, An atlas of physical human–robot interaction, *Mech. Mach. Theory* 43 (3) (2008) 253–270.
- [18] A. Dietrich, C. Ott, Hierarchical impedance-based tracking control of kinematically redundant robots, *IEEE Trans. Robot.* 36 (1) (2019) 204–221.
- [19] J. Vogel, D. Leidner, A. Hagengruber, M. Panzirsch, B. Bauml, M. Denninger, U. Hillenbrand, L. Suchenwirth, P. Schmaus, M. Sewtz, et al., An ecosystem for heterogeneous robotic assistants in caregiving: Core functionalities and use cases, *IEEE Robot. Autom. Mag.* 28 (3) (2020) 12–28.
- [20] M. Iskandar, O. Eiberger, A. Albu-Schäffer, A. De Luca, A. Dietrich, Collision detection, identification, and localization on the dlr sara robot with sensing redundancy, in: 2021 IEEE International Conference on Robotics and Automation, ICRA, IEEE, 2021, pp. 3111–3117.
- [21] P. Aivaliotis, Z. Arkouli, K. Georgoulas, S. Makris, Degradation curves integration in physics-based models: Towards the predictive maintenance of industrial robots, *Robot. Comput.-Integr. Manuf.* 71 (2021) 102177, <http://dx.doi.org/10.1016/j.rcim.2021.102177>.
- [22] P. Aivaliotis, Z. Arkouli, D. Kaliakatsos-Georgopoulos, S. Makris, Prediction assessment methodology for maintenance applications in manufacturing, *Procedia CIRP* 104 (2021) 1494–1499, <http://dx.doi.org/10.1016/j.procir.2021.11.252>, 54th CIRP CMS 2021 - Towards Digitalized Manufacturing 4.0.
- [23] D.D. Rizos, S.D. Fassois, Friction identification based upon the LuGre and maxwell slip models, *IFAC Proc. Vol.* 38 (1) (2005) 548–553, <http://dx.doi.org/10.3182/20050703-6-CZ-1902.00092>, 16th IFAC World Congress.
- [24] M. Linderoth, A. Stolt, A. Robertsson, R. Johansson, Robotic force estimation using motor torques and modeling of low velocity friction disturbances, in: 2013 IEEE/RSJ International Conference on Intelligent Robots and Systems, 2013, pp. 3550–3556, <http://dx.doi.org/10.1109/IROS.2013.6696862>.
- [25] Z.A. Khan, V. Chacko, H. Nazir, A review of friction models in interacting joints for durability design, *Friction* 5 (2017) 1–22.
- [26] R. Hao, J. Wang, J. Zhao, S. Wang, Observer-based robust control of 6-DOF parallel electrical manipulator with fast friction estimation, *IEEE Trans. Autom. Sci. Eng.* 13 (3) (2015) 1399–1408.
- [27] A.C. Bittencourt, S. Gunnarsson, Static friction in a robot joint—modeling and identification of load and temperature effects, 2012.
- [28] A.C. Bittencourt, P. Axelsson, Modeling and experiment design for identification of wear in a robot joint under load and temperature uncertainties based on friction data, *IEEE/ASME Trans. Mechatronics* 19 (5) (2013) 1694–1706.
- [29] A. Krizhevsky, I. Sutskever, G.E. Hinton, ImageNet classification with deep convolutional neural networks, in: F. Pereira, C. Burges, L. Bottou, K. Weinberger (Eds.), *Advances in Neural Information Processing Systems*, Vol. 25, Curran Associates, Inc., 2012.
- [30] D. Silver, A. Huang, C. Maddison, A. Guez, L. Sifre, G. Dieci, J. Schrittwieser, I. Antonoglou, V. Panneershelvam, M. Lanctot, S. Dieleman, D. Grewe, J. Nham, N. Kalchbrenner, I. Sutskever, T. Lillicrap, M. Leach, K. Kavukcuoglu, T. Graepel, D. Hassabis, Mastering the game of go with deep neural networks and tree search, *Nature* 529 (2016) 484–489, <http://dx.doi.org/10.1038/nature16961>.
- [31] A. Fawzi, M. Balog, A. Huang, T. Hubert, B. Romera-Paredes, M. Barekatin, A. Novikov, F. Ruiz, J. Schrittwieser, G. Swirszcz, D. Silver, D. Hassabis, P. Kohli, Discovering faster matrix multiplication algorithms with reinforcement learning, *Nature* 610 (2022) 47–53, <http://dx.doi.org/10.1038/s41586-022-05172-4>.
- [32] J.M. Jumper, R. Evans, A. Pritzel, T. Green, M. Figurnov, O. Ronneberger, K. Tunyasuvunakool, R. Bates, A. Zidek, A. Potapenko, A. Bridgland, C. Meyer, S.A.A. Kohl, A. Ballard, A. Cowie, B. Romera-Paredes, S. Nikolov, R. Jain, J. Adler, T. Back, S. Petersen, D.A. Reiman, E. Clancy, M. Zielinski, M. Steinegger, M. Pacholska, T. Berghammer, S. Bodenstein, D. Silver, O. Vinyals, A.W. Senior, K. Kavukcuoglu, P. Kohli, D. Hassabis, Highly accurate protein structure prediction with AlphaFold, *Nature* 596 (2021) 583–589.
- [33] R. Selmic, F. Lewis, Neural-network approximation of piecewise continuous functions: application to friction compensation, *IEEE Trans. Neural Netw.* 13 (3) (2002) 745–751, <http://dx.doi.org/10.1109/TNN.2002.1000141>.
- [34] M. Ciliz, M. Tomizuka, Neural network based friction compensation in motion control, *Electron. Lett.* 40 (2004) 752–753, <http://dx.doi.org/10.1049/el:20040500>.
- [35] M. Kemal Ciliz, M. Tomizuka, Friction modelling and compensation for motion control using hybrid neural network models, *Eng. Appl. Artif. Intell.* 20 (7) (2007) 898–911, <http://dx.doi.org/10.1016/j.engappai.2006.12.007>.
- [36] S. Huang, K.K. Tan, Intelligent friction modeling and compensation using neural network approximations, *IEEE Trans. Ind. Electron.* 59 (8) (2012) 3342–3349, <http://dx.doi.org/10.1109/TIE.2011.2160509>.
- [37] K. Guo, Y. Pan, H. Yu, Composite learning robot control with friction compensation: A neural network-based approach, *IEEE Trans. Ind. Electron.* 66 (10) (2019) 7841–7851, <http://dx.doi.org/10.1109/TIE.2018.2886763>.
- [38] N. Hirose, R. Tajima, Modeling of rolling friction by recurrent neural network using LSTM, in: 2017 IEEE International Conference on Robotics and Automation, ICRA, 2017, pp. 6471–6478, <http://dx.doi.org/10.1109/ICRA.2017.7989764>.
- [39] X. Tu, Y. Zhou, P. Zhao, X. Cheng, Modeling the static friction in a robot joint by genetically optimized BP neural network, *J. Intell. Robot. Syst.* 94 (2019) 29–41.
- [40] S. Liu, L. Wang, X.V. Wang, Sensorless force estimation for industrial robots using disturbance observer and neural learning of friction approximation, *Robot. Comput.-Integr. Manuf.* 71 (2021) 102168, <http://dx.doi.org/10.1016/j.rcim.2021.102168>.
- [41] B. Armstrong-Helouvry, *Control of Machines with Friction*, Vol. 128, Springer Science & Business Media, 2012.
- [42] M. Iskandar, S. Wolf, Dynamic friction model with thermal and load dependency: modeling, compensation, and external force estimation, in: 2019 International Conference on Robotics and Automation, ICRA, IEEE, 2019, pp. 7367–7373.
- [43] S. Wolf, M. Iskandar, Extending a dynamic friction model with nonlinear viscous and thermal dependency for a motor and harmonic drive gear, in: 2018 IEEE International Conference on Robotics and Automation, ICRA, IEEE, 2018, pp. 783–790.
- [44] C. Canudas de Wit, H. Olsson, K. Astrom, P. Lischinsky, A new model for control of systems with friction, *IEEE Trans. Automat. Control* 40 (3) (1995) 419–425, <http://dx.doi.org/10.1109/9.376053>.
- [45] K. Johansson, C. Canudas-De-Wit, Revisiting the LuGre friction model, *IEEE Control Syst. Mag.* 28 (6) (2008) 101–114.
- [46] V. Vapnik, *The Nature of Statistical Learning Theory*, Springer, New York, 2000.
- [47] M. Abayebas, F. Yumbla, J.-S. Yi, W. Lee, J. Park, H. Moon, Passivity guaranteed dynamic friction model with temperature and load correction: Modeling and compensation for collaborative industrial robot, *IEEE Access* PP (2021) 1, <http://dx.doi.org/10.1109/ACCESS.2021.3076308>.
- [48] D.P. Kingma, J. Ba, Adam: A method for stochastic optimization, in: Y. Bengio, Y. LeCun (Eds.), *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, 2015.