Contents lists available at ScienceDirect

Robotics and Computer-Integrated Manufacturing

journal homepage: www.elsevier.com/locate/rcim

Full length article

Online motion accuracy compensation of industrial servomechanisms using machine learning approaches

Pietro Bilancia [a,*], Alberto Locatelli [a], Alessio Tutarini [b], Mirko Mucciarini [c], Manuel Iori [a], Marcello Pellicciari [a]

[a] *Department of Sciences and Methods for Engineering, University of Modena and Reggio Emilia, Reggio Emilia, Italy*
[b] *Intermech MO.RE., University of Modena and Reggio Emilia, Modena, Italy*
[c] *Department of Economics "Marco Biagi", University of Modena and Reggio Emilia, Modena, Italy*

ARTICLE INFO

Dataset link: https://data.mendeley.com/datasets/g8mrry54j8/1

Keywords:
Servomechanism
Transmission error
Machine learning
Predictive modeling
Compensation approach
Test rig

ABSTRACT

This paper addresses the crucial aspect of position error modeling and compensation in industrial servomechanisms with the aim to achieve accurate control and high-performance operation in industrial robots and automated production systems. The inherent complexity and nonlinear behavior of these modules, usually consisting of a servomotor and a speed reducer, often challenge traditional analytical modeling approaches. In response, the study extensively explores the design and implementation of Machine Learning (ML) algorithms to obtain a comprehensive model of the Transmission Error (TE) in rotating vector reducers, which is a main source of robot motion accuracy errors. The ML models are trained with experimental data obtained from a special purpose test rig, where the reducer is tested under different combinations of input speed, applied load and oil temperature. In the second part of the work, the resulting predictive model, tailored to capture the intricate dynamics of the analyzed reducer, is imported into a programmable logic controller to enable online compensation strategies during the execution of custom motion profiles. Experimental tests are conducted using two distinct motion profiles: one generated with a cycloidal law, typical of industrial machinery, and the other extrapolated from the joints of an industrial robot during a pick-and-place task. The results demonstrate the effectiveness of the proposed approach, enabling accurate prediction and substantial reductions (over 90%) in the overall reducer TE through the implemented predictive model.

1. Introduction

Nowadays, servomechanisms are key components utilized across a wide array of industrial sectors due to their impressive motion dynamic performance and the possibility of easily reconfiguring their tasks, allowing for the rapid adaptation of overall production systems to new requests and formats [1]. They are extensively applied to actuate robotic systems and machine tools but also in various manufacturing processes such as textile, printing, injection molding, metal forming, and food processing [2]. The increasing product quality standards demand to improve the motion accuracy of many manufacturing systems, such as robotic precision machining and assembly [3], metrology equipment [4], Computer Numerical Control (CNC) machines [5], and laser cutting machines [6]. Despite this, commercial servomechanisms typically present various inherent problems that prevent them from meeting strict requirements in terms of position accuracy. With reference to Fig. 1, the differences between the reference position (θ_{ref}) and the effective position obtained at the end-effector (θ_{out}) are caused by:

- the closed-loop control system, namely the non-optimal control law and parameters implemented in the drive systems, which leads to $\theta_{ref} \neq \theta_{mot}$;
- the intermediate mechanical couplings and interfaces (if present), which typically introduce elasticity in the transmission, resulting in $\theta_{mot} \neq \theta_{in}$;
- the speed reducer (installed to better suit the motion and torque requirements of the specific application), which inevitably introduces a Transmission Error (TE) and leads to $\theta_{in} \neq \tau\theta_{out}$ (being τ the reduction ratio).

The study and compensation of the contributions from the servomotor control system and the mechanical connections can potentially be addressed through detailed theoretical and simulation-based analyses, as respectively documented in Refs. [7,8] and [9,10], where identification and tuning procedures are developed and effectively applied to industrial equipment. However, modeling the TE introduced by speed

[*] Corresponding author.
E-mail address: pietro.bilancia@unimore.it (P. Bilancia).

https://doi.org/10.1016/j.rcim.2024.102838
Received 19 July 2024; Accepted 19 July 2024
Available online 29 July 2024

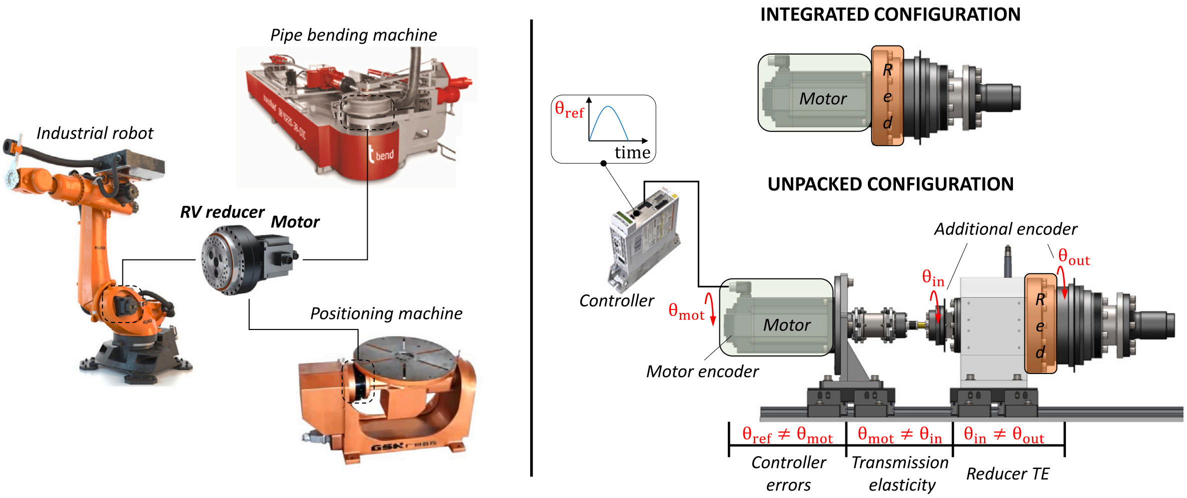


Fig. 1. Industrial applications and mechanical configurations of RV reducers.

reducers can be particularly complex [11,12]. This is primarily due to the intricate nature of the reducers assemblies, comprising numerous parts and mutual complex contact interactions, whose nature may not be perfectly predictable since strictly related to part manufacturing (e.g., tooth shape [13] and tolerances [14]). The complexity intensifies with compact solutions like Rotating Vectors (RV) reducers, which combine a planetary stage with a cycloidal stage to achieve higher reduction ratios (typically ranging from 30 to 300, as visible from the manufacturers catalogues [15,16]).

Generally, the dynamic behavior of speed reducers varies based on their kinematic architecture (see [17] for a review) and a multitude of operating parameters, including speed, resistant torque, lubricant temperature, and the state of degradation, as assessed in Refs. [10,18]. The intricate interplay of these factors necessitates a comprehensive approach when attempting to model and understand their effect on the TE. Purely theoretical models, as the ones proposed in Refs. [13,19], may not adequately capture the parameter sensitivity essential for predicting and compensating for these nonlinear effects in real-world applications. Also, additional challenges emerge from the numerous geometric and functional parameters of the selected commercial reducer remaining hidden to the user, limiting the applicability of the provided formulae. Consequently, recent studies have increasingly employed specialized testing equipment to enrich behavioral models and gain insights into these complex interactions [10,20–23]. Although the referenced works have made significant advances into the reducers dynamic characterization, the reported investigations typically consider a reduced number of parameters (e.g. combination of speed and applied load, or wear, excluding the oil temperature).

Experiments conducted on opportunely instrumented test rigs yield extensive data and offer a more comprehensive understanding of the system under investigation compared to direct measurements on industrial machines [4]. Nevertheless, mapping the reducer TE across a broad operational domain (i.e. considering 3 to 4 input parameters) can become time-consuming. This is attributed to the high number of required experiments (even with a medium number of levels assigned to each parameter), influenced by the chosen fitting method aimed at building accurate empirical relations resorting to meta-modeling techniques (the interested reader should refer to [24]). In this context, Machine Learning (ML) has emerged as a valuable tool to achieve a good balance between the number of experiments and the model accuracy, as documented in Refs. [25,26]. ML algorithms can analyze historical data, identify patterns, and predict system behavior, making them suitable for modeling complex, nonlinear dynamic systems. In the literature, ML has been employed to develop predictive models that can compensate for errors, improve stability, and continuously optimize the performance of control systems [27] and industrial servomechanisms [28]. ML has also been applied to predict the position accuracy errors of industrial robots in their workspace [29–31] and the failures during robotic tasks [32], or to perform optimal trajectory generation and path error compensation in both robots [33] and CNC machines [34].

Despite these examples have set the stage for the future integration of ML in industry and a well-documented repository of ML models already exists (see e.g. [35]), the current state-of-the-art lacks comprehensive discussions offering rule-of-thumb guidelines for effective selection and implementation of ML models on commercial industrial controllers. Also, ML predictive models are typically utilized within offline compensation frameworks, where corrections are calculated and then applied to the original motion laws prior to their implementation and execution on the control system [36–38]. In contrast, online compensation frameworks usually involve a sensor-based closed-loop controller as in the works presented in Refs. [39,40], where the robot paths are corrected in real-time based on the position feedback registered via a laser tracker. Striking a balance between both, online compensation with model-based corrections emerges as a potential strategy that allows for real-time corrections without the need for manual preprocessing of the motion law or precise tuning the control system. This approach not only enhances the operational efficiency in modern automated systems [41] but also circumvents the necessity for expensive and delicate high-class sensor systems (e.g. laser trackers and interferometers), thereby eliminating the need for advanced personnel skills for their setup, tuning, and maintenance.

Within this context, the research presented in this paper delves into the critical domain of position error modeling and compensation in industrial servomechanisms, aiming to enhance control precision and operational performance in automated production systems. In particular, with the aim to bridge the existing gaps in the literature, the novel contributions of this paper are as follows:

1. **Engineering methods and tools** to conduct structured experimental parametric studies on RV reducers via a purposely designed test rig apparatus.
2. **Sensitivity analysis** of the motion performance of RV reducers by evaluating the effect of the main operating parameters (speed, applied torque and lubricant temperature) on their TE functions, with particular focus on position-dependent phenomena, namely events happening a specific number of times per revolution (θ_{out}) and therefore associated with specific spectral components at each rotating speed.

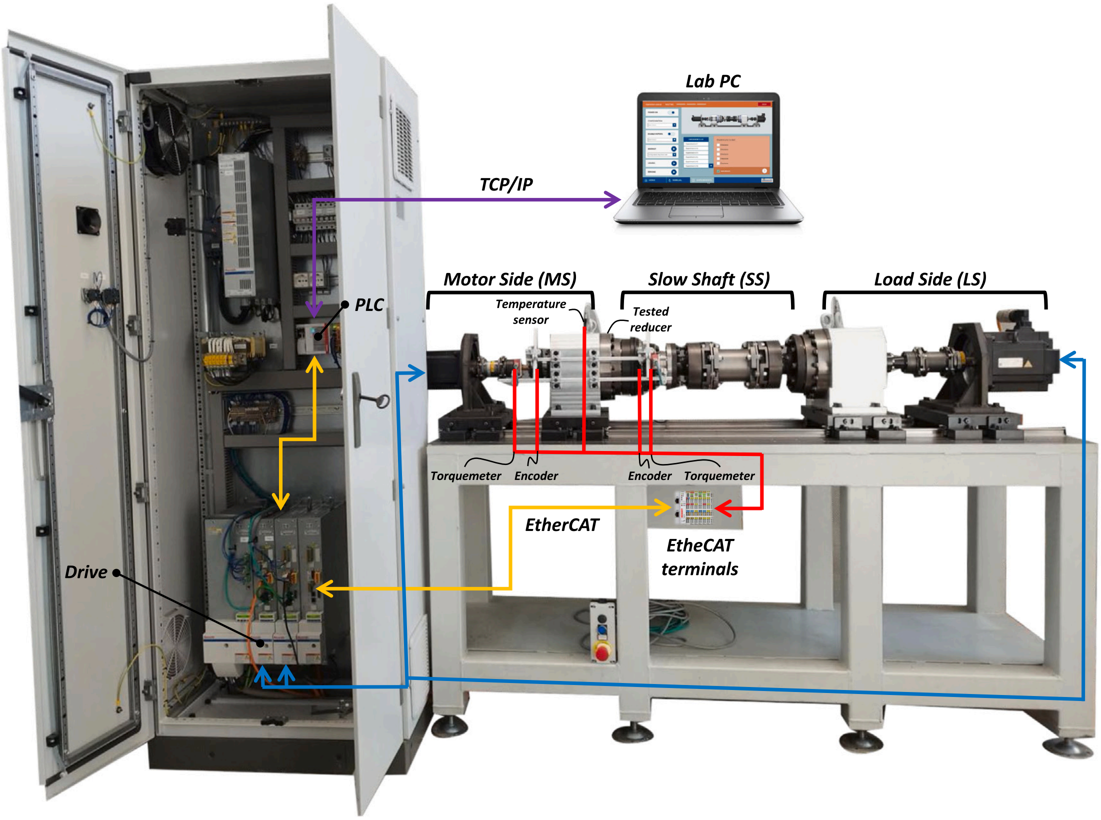


Fig. 2. Test rig schematic: functional modules and main connections.

3. **Implementation of predictive ML models** of the TE in RV reducers that efficiently correlate the reducer operating parameters with the forward and backward TE functions over an entire revolution of the output shaft (θ_{out}). The ML models are developed using Python and are trained with experimental data obtained from the rig.
4. **Integration of the predictive ML models** into a Programmable Logic Controller (PLC) to perform real-time online compensation during the execution of customized motion profiles with model-based corrections.
5. **Open source dataset** related to the performed TE measurements and obtained ML models, shared to ease further developments.

The rest of the paper follows this structure: Section 2 describes the experimental equipment and discusses the influences of the main operating parameters on the measured TE functions; Section 3 outlines the theoretical aspects and thoroughly explores the design of the ML algorithms aimed at acquiring a comprehensive model of the reducer TE; Section 4 reports on the integration of the ML model into a commercial PLC by Beckhoff; Section 5 presents and discusses the experimental tests conducted to compensate custom motion profiles and validate the proposed approach. The concluding remarks are given in Section 6.

2. Experimental setup and data collection

This section provides a comprehensive overview of the employed test rig apparatus, detailing its mechanical architecture, primary hardware components, and acquisition system. Subsequently, it presents a preliminary qualitative analysis of the TE spectral characteristics of RV reducers, along with a description of the testing methodologies employed for collecting ML training data.

2.1. Equipment

The test rig setup, shown in Fig. 2, consists of two interconnected servomechanisms linked by a central shaft. The primary servomechanism, identified as the system under examination, integrates a BOSCH servomotor paired with a Nabtesco RV 160N reducer (reduction ratio $\tau = 81$, 40 pins and 39 teeth on the cycloidal gear). The secondary servomechanism, comprising a BOSCH servomotor and a similar reducer with a reduction ratio of 156, operates as the active loading system. Designed for time-varying torque profiles and velocities, this loading system faithfully replicates the dynamic conditions encountered by the examined servomechanism in practical service scenarios, such as the industrial environments visible in Fig. 1. Both servomechanisms are under the control of a PLC provided by Beckhoff. The PLC facilitates synchronized communication with the drives, responsible for the control and regulation of the power supplied to the servomotors. The communication among the devices is established through an EtherCAT fieldbus. In this setup, the PLC assumes the role of the master, whereas the I/O devices function as slaves. The rig functioning is programmed and managed from a standard PC utilizing a special user interface implemented with Beckhoff TwinCAT HMI and connected to the PLC via TCP/IP. For an in-depth understanding of the design considerations inherent to this rig prototype, the interested readers may consult Refs. [10,42].

As it can be seen from Fig. 2, the presented rig architecture consists of three main modules, namely the Motor Side (MS), the Slow Shaft (SS) and the Load Side (LS). The MS and LS represent the mechanical transmissions at the input side of the tested reducer and of the secondary reducer respectively, whereas the SS is the central connecting shaft. To properly assess the performance of the first (left) reducer, additional optical encoders by Renishaw and torquemeter by Manner are installed on both the MS and SS. Additionally, a temperature sensor is incorporated in the reducer case to monitor the oil temperature variations, which influence the dynamic behavior of the reducer, as documented in Refs. [10,43], where notable changes in terms of position errors can be observed for both the single joint and the entire serial robotic system. To respect the measurement limits posed by the installed sensors, maximum torque values of 50 N m and 2000 N m for the MS and SS are considered in this work. In the same way, considering the speed limits of the secondary servomotor (4000 rpm) and the reduction ratios of both reducers (81 and 156), the maximum operated speed at MS is fixed to 1800 rpm.

2.2. TE function analysis

The accuracy assessment of the RV reducer is conducted focusing on the evaluation of the TE function, which provides crucial insights into its dynamic performance. By reading the angular positions at the reducer input (θ_{in}) and output (θ_{out}) shafts with the encoders installed on the MS and SS respectively (see Figs. 1 and 2), the TE is simply obtained at each point as:

$$TE(\theta_{out}) = \theta_{out} - \frac{\theta_{in}}{\tau} \tag{1}$$

For a comprehensive understanding of the reducer behavior, the TE functions are normally measured for both forward and backward motions, resulting in two distinct curves (TE_f and TE_b). From the findings of prior studies dealing with the analysis of RV reducer [10,20,44], and the TE plots depicted in Fig. 3, the dependency of TE from the input speed (ω_{in}), external applied load (M_{out}), and lubricant temperature (T_l) becomes evident. In particular, the most significant harmonics in the related spectrum correspond to specific physical phenomena that impact the performance of the reducer mechanism. A notable observation derived from Fig. 3 is that these peaks always occur at the following normalized frequencies (i.e. $f/f_{0,\mathrm{out}}$, being $f_{0,\mathrm{out}}$ the rotational speed of the output shaft expressed in rps):

- Component 0: associated with backlash and elastic torsional shaft deformation when subject to high external loads or internal friction loads (see, e.g., the high dependency from M_{out} in Fig. 3).
- Component 1: arises from the deviation of the output shaft.
- Component 3: possibly attributed to misalignment between the planetary gear shaft and the cycloidal disk, occurring three times in one complete revolution of the output shaft.
- Component 39: linked to errors in the cycloidal gear teeth (influenced by M_{out} as it causes their local elastic deformation).
- Component 40: arises from errors in the pins.
- Component 78: 2nd harmonic of the errors in the cycloidal gear teeth. For higher M_{out} values, it impacts more than the component 39.
- Component 81: reflects misalignment in the fast shaft, which manifests 81 times in one revolution of the output shaft.
- Component 156: 4th harmonic of the errors in the cycloidal gear teeth.
- Component 162: 2nd harmonic of the misalignment in the fast shaft.
- Component 240: 6th harmonic of the pins.

Hence, according to Ref. [45], the reducer TE function can be mathematically expressed as the sum of the most significant harmonics

$$TE\_(\theta_{out}, \omega_{in}, T_l, M_{out}) = \sum_{k\in P} A_{\_,k} \cos(k\theta_{out} + \phi_{\_,k}) \tag{2}$$

being P the set of selected harmonics, and $A_{\_,k}$ and $\phi_{\_,k}$ the amplitude and phase shift of harmonic k, respectively. The subscript _ can be f or b, indicating forward or backward rotations. Therefore, by knowing $A_{\_,k}$ and $\phi_{\_,k}$ for each harmonic $k \in P$ and for each combination of ω_{in}, M_{out} and T_l, one could potentially obtain the TE functions of a specific reducer model in all its working range.

2.3. Data collection

Based on the TE characteristics outlined earlier and with the objective of constructing a comprehensive TE model for the reducer using the relation expressed in Eq. (2), an extensive dataset has been compiled for the mounted RV reducer to train ML algorithms. The aim is to predict the values of $A_{\_,k}$ and $\phi_{\_,k}$ within the operational range of the reducer to facilitate rapid TE calculations. In the frequency domain, these properties inherently encapsulate the complex dynamics of the reducer. The experimental study is organized as a three-dimensional grid, encompassing all possible combinations of the following parameter values:

- $\omega_{in} \in \{100, 200, \dots, 1800\}$ rpm (18 levels)
- $M_{out} \in \{0, 100, \dots, 1800\}$ N m (19 levels)
- $T_l \in \{25, 30, 35\}$ °C (3 levels)

A total of 1026 experiments have been conducted for both the forward and backward rotations. The collected data underwent comprehensive analysis and served as the foundation for designing the ML model, as it will be explained in Section 3.

2.4. Experimental procedure

Due to the high number of experiments, a script enabling their automated execution has been created and loaded on the PLC. The script operates through a state machine, performing the following sequence of steps:

- *Preliminary check of input parameters*, specifically speed on MS and LS and torque on SS, to ensure they meet the limits specified in Section 2.1.
- *Check of encoder state* to confirm that both the additional optical encoders are operating correctly, providing accurate measurements of θ_{in} and θ_{out}.
- *Warm-up phase* by imposing a continuous motion at constant speed on the MS until the oil temperature in the MS reducer (T_l) reaches the required level. During this phase, the LS motor is enabled with a zero torque command ($M_{LS} = 0$ N m) to reduce the workload on the MS motor responsible for driving the entire transmission.
- *Homing procedure* to position the examined reducer at its zero angle at the output side ($\theta_{out} = 0°$). Slight corrections are subsequently made with the MS motor to position the input side in the middle of the angular backlash interval. The MS is kept active to maintain such position and a null torque is commanded on the SS ($M_{out} = 0$ N m) to unload the shaft and facilitate the operation. The procedure concludes with the software zeroing of both encoders.
- *Experiment execution* by driving the MS motor at a constant speed. The SS torque is incrementally increased to reach the input-defined value. When the SS encoder crosses $\theta_{out} = 0°$, the *Data-valid* signal is set to TRUE and then again to FALSE after a complete revolution. This step is crucial for the post-processing in order to map the TE for $0° \leq \theta_{out} \leq 360°$.

This experiment is repeated for both rotational directions, enabling the evaluation of TE_f and TE_b. In the backward experiment, to maintain the same angular reference, the initialization process (including *Warm-up* and *Homing*) is not replicated. Each experiment took about 5 to 10 min to complete, leading to an overall time of about 171 h to run all the 1026 forward and backward datasets.

3. A ML framework for the TE prediction

This section presents a general purpose ML framework specifically developed for estimating the TE in RV reducers. The framework incorporates several ML regression models trained on the empirical dataset

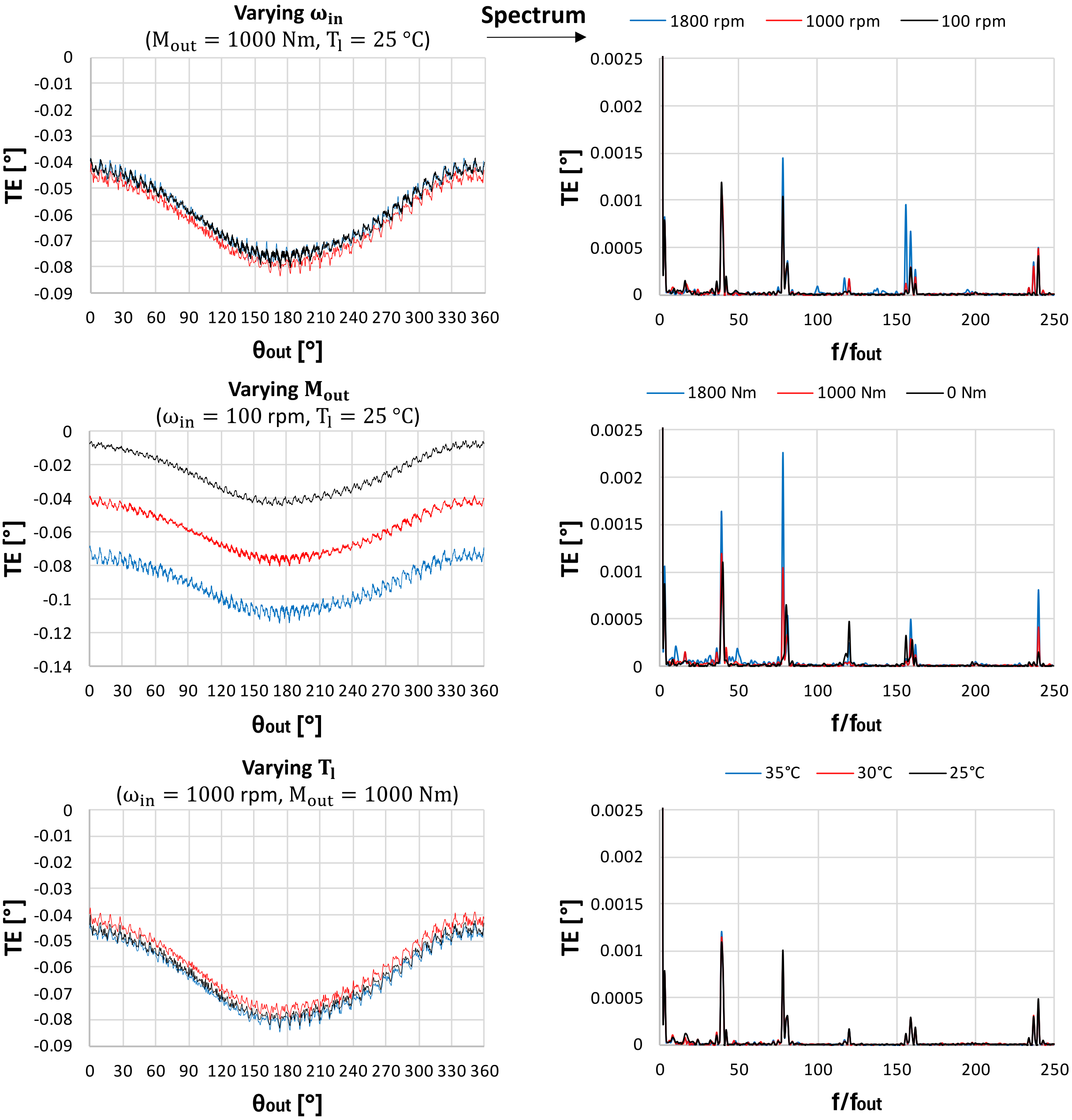


Fig. 3. Transmission performance of a Nabtesco RV reducer: TE function expressed over an entire revolution of the output shaft and related spectrum.

presented in Section 2.3. The models are designed and implemented to accurately estimate $A_{\_,k}$ and $\phi_{\_,k}$ according to the specific input data (i.e., speed, applied load, and oil temperature). Once these values are obtained, Eq. (2) is utilized to compute the TE.

As highlighted in Section 2.2, the relationship between specific harmonics of Eq. (2) and their associated physical mechanisms are known. However, a comprehensive characterization of $A_{\_,k}$ and $\phi_{\_,k}$ in terms of ω_{in}, M_{out} and T_l remains unexplored. To fill this gap in the literature, this section proposes a data-driven approach by using supervised ML algorithms capable of making fast predictions based only on historical data. This choice is motivated by the complexity associated with predicting quantities that cannot be easily computed using analytic formulas. Adopting a data-driven approach, the ML models, obtained by training ML algorithms, effectively capture the inherent nonlinearity and complexity of the system and can be easily retrained with new data to adapt to evolving system conditions. Another crucial characteristic that ensures the ML models are suited to address the compensation problem is their capacity to make fast predictions. Indeed, these models can be easily managed by the PLC for real-time adjustments using specific functionalities, as it will be shown in Section 4.

3.1. Methodology

According to the flowchart depicted in Fig. 4, the proposed methodology begins with data collection, followed by data analysis. The spectral analysis leads to the definition of set P, which contains the selected harmonic orders. Then, for each $k \in P$, the task of forecasting $A_{\_,k}$ is defined as a regression problem where the objective is to predict $A_{\_,k}$ using a function $\widetilde{A}_{\_,k}$ of a 3-dimensional set of features. To face this problem, a data-driven approach is employed through the use of supervised ML algorithms to create ML models for $\widetilde{A}_{\_,k}$. In this phase, a training dataset $\mathcal{D}_{\text{train},A_{\_,k}} = \{(x^i, A^i_{\_,k})\}_{i=1}^m$ is provided (for further details see Section 3.3). Each pair of $\mathcal{D}_{\text{train},k}$ includes a 3-dimensional input feature vector $x^i = (\omega^i_{in}, T^i_l, M^i_{out})$ and the corresponding desired

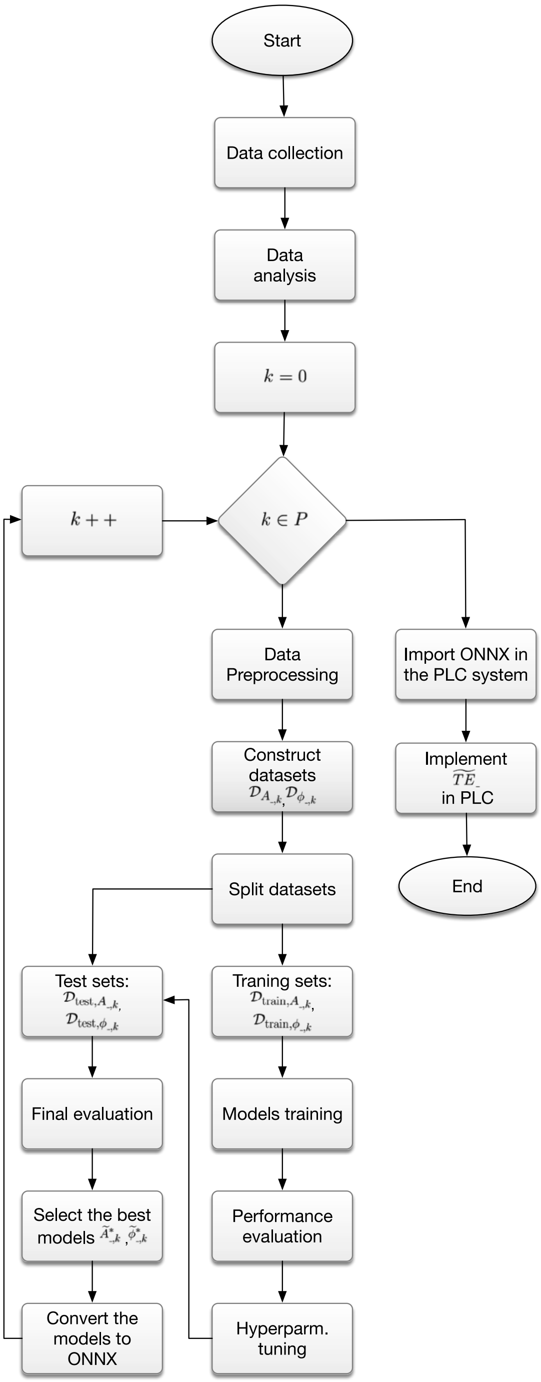

Start
Data collection
Data analysis
$k = 0$
$k++$
$k \in P$
Data Preprocessing
Import ONNX in the PLC system
Construct datasets $\mathcal{D}_{A_{\_,k}}, \mathcal{D}_{\phi_{\_,k}}$
Implement $\widetilde{TE}_{\_}$ in PLC
Split datasets
End
Test sets: $\mathcal{D}_{\text{test},A_{\_,k}}$, $\mathcal{D}_{\text{test},\phi_{\_,k}}$
Traning sets: $\mathcal{D}_{\text{train},A_{\_,k}}$, $\mathcal{D}_{\text{train},\phi_{\_,k}}$
Final evaluation
Models training
Select the best models $\widetilde{A}^*_{\_,k}$,$\widetilde{\phi}^*_{\_,k}$
Performance evaluation
Convert the models to ONNX
Hyperparm. tuning

Fig. 4. Flowchart of the designed ML training process.

output $A^i_{\_,k}$. Thus, the learning task is to find a predictor function $\widetilde{A}_{\_,k}$, belonging to a certain family of ML models, such that, for each input x^i, the predicted output $\widetilde{A}_{\_,k}(x^i)$ is as close as possible to the provided target $A^i_{\_,k}$. In Section 3.4, several families of ML models are explored and the results are used to identify the best predictor function $\widetilde{A}^*_{\_,k}$. Then, in Section 3.5, the final unbiased generalization error of the selected model is computed using a provided test set $\mathcal{D}_{\text{test},A_{\_,k}}$. Finally, $\widetilde{A}^*_{\_,k}$ is converted into the standardized Open Neural Network Exchange (ONNX) format, to be imported into the PLC. The same methodology employed to forecast $A_{\_,k}$ is also applied to predict $\phi_{\_,k}$. After importing the selected ML models into the PLC, based on Eq. (2), the implemented mathematical formulation for calculating TE becomes:

$$\widetilde{TE}_{\_}(\theta_{out},\omega_{in},T_l,M_{out}) = \sum_{k\in P} \widetilde{A}^*_{\_,k}\cos(k\theta_{out}+\widetilde{\phi}^*_{\_,k}) \qquad (3)$$

3.2. Data description

The initial dataset $\mathcal{D}$ originates from the experimental campaign described in Section 2.3. During the experiments, the signal acquisition is handled with a Beckhoff CX5140 PLC. For each specific combination of ω_{in}, T_l, and M_{out}, the relative test-generated signal is acquired using encoders operating at a sampling frequency of 4000 Hz. This signal is then processed using Eq. (1) to compute the value of TE. The results are subsequently stored in a Comma-Separated Values (CSV) file, containing four columns of data. The first and second columns of the file report the angular positions θ_{out}, during a full forward motion (from 0° to 360°), and the corresponding value $TE_f(\theta_{out},\omega_{in},T_l,M_{out})$, respectively. Similarly, the third and fourth columns show the angular positions θ_{out}, during the backward motion (from 360° back to 0°), and the corresponding value $TE_b(\theta_{out},\omega_{in},T_l,M_{out})$, respectively. Thus, the initial dataset $\mathcal{D}$ consists of a total of $m = 1026$ CSV files, each related to a specific experiment. During the experiments, the identification of anomalies in data points prompted the repetition of tests, ensuring the high quality of $\mathcal{D}$. Consequently, no data-cleaning phase has been implemented on $\mathcal{D}$.

3.3. Data preprocessing phase

For each harmonic $k \in P$ and for each experiment $i \in \mathcal{D}$ the coefficients $A^i_{\_,k}$ and $\phi^i_{\_,k}$ have been extrapolated via Fast Fourier Transform. The obtained coefficients have been used to construct the datasets $\mathcal{D}_{A_{\_,k}} = \{(x^i, A^i_{\_,k})\}_{i=1}^{1026}$ and $\mathcal{D}_{\phi_{\_,k}} = \{(x^i, \phi^i_{\_,k})\}_{i=1}^{1026}$, respectively. Following the dataset creation process, $\mathcal{D}_{A_{\_,k}}$ has been randomly partitioned into two subsets: the training set $\mathcal{D}_{\text{train},A_{\_,k}}$, containing 80% of the signals and used for the model fitting phase, and a test set $\mathcal{D}_{\text{test},A_{\_,k}}$, which contains the remaining 20% of the signals and is used for model evaluations. A similar partition has been applied to $\mathcal{D}_{\phi_{\_,k}}$, resulting in the creation of the training dataset $\mathcal{D}_{\text{train},\phi_{\_,k}}$ and the test set $\mathcal{D}_{\text{test},\phi_{\_,k}}$.

3.4. ML models

Although the existing literature lacks any application of ML models for TE prediction in RV reducers, various types of ML models have been utilized for predicting the position accuracy errors of industrial robots in their workspace. In this context, Neural Networks (NNs) have been widely used due to their advanced learning abilities and flexibility (see e.g. [46–48]). Recently, some advanced models based on deep learning have also been successfully applied [29,36]. However, as highlighted in [49], these models operate on empirical risk minimization, which means that they perform well with a large amount of training data. On the other hand, when dealing with limited training data, tailored variants of Support Vector Machine (SVM) have demonstrated good results in terms of accuracy [49,50], as well as Extreme Learning Machine (ELM) [38,51].

This section offers an overview of the ML techniques selected to address the regression problem described in Section 3.1. Since the main objective of this article is to present a flexible ML framework operating with a Beckhoff CX5140 PLC for the prediction and compensation of TE, the choice of ML models is limited to those compatible with integration into the TwinCAT environment. According to the Beckhoff documentation (available at https://infosys.beckhoff.com), TwinCAT provides support only for the following ML regression models:

- Support Vector Machine (SVM) aims to find a hyperplane that optimally separates the data while minimizing the margin of error between the predicted and actual values [52].

- Multi-Layer Perceptron (MLP) is a NN that consists of a stack of three (or more) layers, each consisting of a certain number of neurons. The first layer consists of input variables. In the second layer, named the hidden layer, the output of each neuron is computed as a nonlinear combination of input variables, whose weights are learned during a training phase. The last layer computes the output of the network as a nonlinear combination of the output of the hidden neurons, again with adjustable, learnable weights [53].
- Decision Tree (DT) uses a tree-like structure to recursively split the dataset into subsets based on feature values, enabling the prediction of continuous numeric values by averaging the target variable within each leaf node of the tree [54].
- Extra Tree (ET) is a type of DT that introduces an additional level of randomness in the tree-building process. While a standard DT selects the best splitting feature at each node, ET randomly chooses a feature and a random threshold for each split [55].
- Random Forest (RF) is an ensemble learning method made up of a number of DTs, called estimators, each of which is a weak prediction model with limited predictive power. However, an RF model leverages the diversity of these weak models to produce a more accurate prediction by combining them together and aggregating their individual predictions [56].
- Extremely Randomized Tree (ERT), similarly to RF, fits a number of ETs on various sub-samples of the dataset and uses averaging to improve the predictive accuracy and control overfitting [55].
- Gradient Boosting Machine (GBM) is an ensemble ML model that sequentially combines weak predictive models, typically DTs, to improve overall prediction accuracy. It minimizes errors by adjusting model weights based on the residuals of the previous iterations, resulting in a robust and accurate regression model [57]. TwinCAT provides support for the following three well-known implementations of GBM:
  - Histogram-Based Gradient Boosting Machine (HGBM) differs from GBM by adopting a more efficient approach in splitting feature values when constructing decision trees. In contrast to GBM, HGBM discretizes continuous features into discrete bins and uses these bins to create feature histograms during training. This approach results in faster and more scalable tree construction, making it particularly effective for handling large datasets [58].
  - LightGBM (LGBM) is known for its speed and memory efficiency [59].
  - XGBoost (XGBM) incorporates a variety of regularization techniques, tree-pruning algorithms, and cross-validation support, making it a versatile and effective tool [60].

As already highlighted, the regression problems defined in Section 3.1 are characterized by complex nonlinear relationships between the input features and the target variable. While all described models are capable of describing nonlinear behaviors, their effectiveness in terms of prediction accuracy varies. Consequently, Section 3.6 introduces performance evaluation metrics aimed at analyzing the prediction accuracy of the models.

3.5. *Models evaluation tests*

Computational tests have been conducted to evaluate the performances of the ML models described in Section 3.4. Data preparation and preprocessing, detailed in Sections 3.2 and 3.3, are performed using the `pandas` and `numpy` libraries in Python. The ML models are developed using the open-source `scikit-learn` library [35,61] in Python. The training and the test of the models are executed on a computer equipped with a 2.3 GHz Intel Xeon Gold 6252N processor and 16 GB of memory.

Table 1
Grid search cross-validation on the models.

Model	Hyperparameter	Values
SVR	Regularization param. C	0.01, 0.1, 0.5, **1**, 2, 3
	Kernel function K	linear, **rbf**
	epsilon	$1\cdot 10^{-3}$, $\mathbf{1\cdot 10^{-4}}$, $1\cdot 10^{-5}$, $1\cdot 10^{-6}$
	gamma	$1\cdot 10^{-5}$, $\mathbf{1\cdot 10^{-6}}$, $1\cdot 10^{-7}$
MLP	Activation function	**tanh**, relu
	Early stopping	True, False
	Hidden layer(s) size	(100,), (100, 50), (200,), **(200, 50)**
	Learning rate	**adaptive**, constant
	Solver	sgd, **adam**
RF	N estimators	50, 60, 70, 80, **90**, 100
	Criterion	**squared error**, absolute error
	Max depth	12, 13, **14**, 15, 16, None
	Min samples split	2, **3**, 4, 5, 6
DT	Criterion	**squared error**, absolute error
	Max depth	14, 15, **16**, 17, 18, None
	Min samples split	2, 4, 5, **6**, 7, 8
ET	Criterion	**squared error**, absolute error
	Max depth	14, **15**, 16, 17, 18, None
	Min samples split	2, 3, 4, **5**, 6
ERT	N estimators	20, 40, **60**, 80, 100
	Criterion	**squared error**, absolute error
	Max depth	14, **15**, 16, 17, 18, None
	Min samples split	2, **3**, 4, 5, 6
GBM	N estimators	20, **36**, 52, 68, 84, 100
	Criterion	**squared error**, absolute error
	Max depth	13, **14**, 15, 16, 17, 18,
	Learning rate	0.001, 0.01, **0.1**, 1.0
	Min samples split	10, 12, **14**, 16, 18,
HGBM	Max depth	9, 10, **11**, 12, 13, None
	Learning rate	0.18, 0.19, 0.2, **0.21**, 0.22, 0.1
	Max leaf nodes	24, 25, 26, **27**, 28, 31
LGBM	Max depth	10, 11, **12**, 13, 14, 15
	Learning rate	0.01, 0.1, 0.2, **0.39**, 0.58, 0.77
	Subsample	0.001, 0.01, **0.1**, 0.2, 0.3
XGBM	Max depth	12, 14, **16**, 18, 20
	Colsample by tree	0.6, 0.7, **0.8**, 0.9, 1
	Alpha	0, 0.1, **0.01**, 0.001, 0.0001
	Lambda	1, 10, **20**, 30, 40

The models have been trained on the dataset $\mathcal{D}_{\text{train}}$, which consists of 820 samples. The evaluation of the ML models has been conducted by using a 10-fold cross-validation approach, a procedure that is typically recommended for small datasets [62]. Specifically, the dataset $\mathcal{D}_{\text{train}}$ is randomly divided into ten equally sized segments (also referred to as folds in the literature). Ten distinct models are then trained on nine of these segments, with the remaining segment being used for testing purposes.

Each ML model underwent training using the grid search cross-validation technique, provided by `scikit-learn`, over a specific hyperparameter grid outlined in Table 1 and determined through preliminary experimental trials. Any additional setting is left to the default value as implemented in the `scikit-learn` library. The grid search algorithm generated all the combinations of hyperparameters and identified the best ones, highlighted in bold in Table 1. This fine-tuning facilitates the optimization of each model, customizing them to the specific characteristics of the dataset and problem domain, ensuring optimal performance and smooth integration into the PLC environment. Although this approach requires manual effort, it is preferred over the use of more user-friendly but less adaptable Automated Machine Learning (AutoML) systems [63].

3.6. *Evaluation criteria*

The accuracy of the trained models has been assessed by using two comprehensive statistical indicators: Mean Absolute Error (MAE)

Table 2
$\mathrm{MAE}_{\widetilde{A}_{\_,k}}$ values of the trained ML models.

Model	Harmonics order k									
	0	1	3	39	40	78	81	156	162	240
SVM	$\mathbf{2.3 \cdot 10^{-3}}$	$5.6 \cdot 10^{-5}$	$1.6 \cdot 10^{-4}$	$1.5 \cdot 10^{-4}$	$7.9 \cdot 10^{-5}$	$2.6 \cdot 10^{-4}$	$9.1 \cdot 10^{-5}$	$4.4 \cdot 10^{-4}$	$6.9 \cdot 10^{-4}$	$2.9 \cdot 10^{-4}$
MLP	$9.5 \cdot 10^{-3}$	$6.5 \cdot 10^{-3}$	$6.5 \cdot 10^{-3}$	$5.6 \cdot 10^{-3}$	$6.9 \cdot 10^{-3}$	$7.1 \cdot 10^{-3}$	$7.4 \cdot 10^{-3}$	$6.8 \cdot 10^{-3}$	$8.1 \cdot 10^{-3}$	$5.5 \cdot 10^{-3}$
RF	$3.0 \cdot 10^{-3}$	$\mathbf{2.6 \cdot 10^{-5}}$	$2.0 \cdot 10^{-5}$	$2.9 \cdot 10^{-5}$	$2.6 \cdot 10^{-5}$	$3.8 \cdot 10^{-5}$	$\mathbf{1.1 \cdot 10^{-5}}$	$5.7 \cdot 10^{-5}$	$6.8 \cdot 10^{-5}$	$2.9 \cdot 10^{-5}$
DT	$3.4 \cdot 10^{-3}$	$2.9 \cdot 10^{-5}$	$2.2 \cdot 10^{-5}$	$4.0 \cdot 10^{-5}$	$3.2 \cdot 10^{-5}$	$5.9 \cdot 10^{-5}$	$1.3 \cdot 10^{-5}$	$6.3 \cdot 10^{-5}$	$6.2 \cdot 10^{-5}$	$5.1 \cdot 10^{-5}$
ET	$3.5 \cdot 10^{-3}$	$3.1 \cdot 10^{-5}$	$2.4 \cdot 10^{-5}$	$3.8 \cdot 10^{-5}$	$3.2 \cdot 10^{-5}$	$5.9 \cdot 10^{-5}$	$1.8 \cdot 10^{-5}$	$5.7 \cdot 10^{-5}$	$8.8 \cdot 10^{-5}$	$7.2 \cdot 10^{-5}$
ERT	$3.1 \cdot 10^{-3}$	$2.7 \cdot 10^{-5}$	$2.3 \cdot 10^{-5}$	$2.9 \cdot 10^{-5}$	$\mathbf{2.5 \cdot 10^{-5}}$	$3.8 \cdot 10^{-5}$	$1.2 \cdot 10^{-5}$	$\mathbf{3.9 \cdot 10^{-5}}$	$\mathbf{4.5 \cdot 10^{-5}}$	$\mathbf{2.6 \cdot 10^{-5}}$
GBM	$3.1 \cdot 10^{-3}$	$2.7 \cdot 10^{-5}$	$2.1 \cdot 10^{-5}$	$2.8 \cdot 10^{-5}$	$2.7 \cdot 10^{-5}$	$3.9 \cdot 10^{-5}$	$1.2 \cdot 10^{-5}$	$6.1 \cdot 10^{-5}$	$7.1 \cdot 10^{-5}$	$3.0 \cdot 10^{-5}$
HGBM	$2.4 \cdot 10^{-3}$	$2.7 \cdot 10^{-5}$	$\mathbf{1.7 \cdot 10^{-5}}$	$\mathbf{2.3 \cdot 10^{-5}}$	$2.6 \cdot 10^{-5}$	$\mathbf{2.9 \cdot 10^{-5}}$	$1.2 \cdot 10^{-5}$	$1.0 \cdot 10^{-4}$	$1.7 \cdot 10^{-4}$	$3.5 \cdot 10^{-5}$
XGBM	$2.5 \cdot 10^{-3}$	$5.5 \cdot 10^{-5}$	$8.1 \cdot 10^{-5}$	$1.1 \cdot 10^{-4}$	$6.6 \cdot 10^{-5}$	$1.1 \cdot 10^{-4}$	$4.6 \cdot 10^{-5}$	$2.3 \cdot 10^{-4}$	$2.6 \cdot 10^{-4}$	$1.4 \cdot 10^{-4}$
LGBM	$2.5 \cdot 10^{-3}$	$2.7 \cdot 10^{-5}$	$1.8 \cdot 10^{-5}$	$2.4 \cdot 10^{-5}$	$2.7 \cdot 10^{-5}$	$3.0 \cdot 10^{-5}$	$1.2 \cdot 10^{-5}$	$9.0 \cdot 10^{-5}$	$1.6 \cdot 10^{-4}$	$3.2 \cdot 10^{-5}$

Table 3
$\mathrm{RMSE}_{\widetilde{A}_{\_,k}}$ values of the trained ML models.

Model	Harmonics order k									
	0	1	3	39	40	78	81	156	162	240
SVM	$\mathbf{3.3 \cdot 10^{-3}}$	$7.4 \cdot 10^{-5}$	$1.8 \cdot 10^{-4}$	$1.8 \cdot 10^{-4}$	$9.5 \cdot 10^{-5}$	$3.3 \cdot 10^{-4}$	$1.0 \cdot 10^{-4}$	$8.8 \cdot 10^{-4}$	$2.2 \cdot 10^{-3}$	$4.7 \cdot 10^{-4}$
MLP	$1.4 \cdot 10^{-2}$	$1.2 \cdot 10^{-2}$	$1.2 \cdot 10^{-2}$	$1.0 \cdot 10^{-2}$	$1.4 \cdot 10^{-2}$	$1.3 \cdot 10^{-2}$	$1.5 \cdot 10^{-2}$	$1.3 \cdot 10^{-2}$	$1.6 \cdot 10^{-2}$	$1.0 \cdot 10^{-2}$
RF	$4.1 \cdot 10^{-3}$	$\mathbf{3.5 \cdot 10^{-5}}$	$3.0 \cdot 10^{-5}$	$3.8 \cdot 10^{-5}$	$3.7 \cdot 10^{-5}$	$5.6 \cdot 10^{-5}$	$\mathbf{1.5 \cdot 10^{-5}}$	$1.7 \cdot 10^{-4}$	$2.2 \cdot 10^{-4}$	$5.4 \cdot 10^{-5}$
DT	$4.9 \cdot 10^{-3}$	$4.0 \cdot 10^{-5}$	$3.3 \cdot 10^{-5}$	$5.3 \cdot 10^{-5}$	$4.5 \cdot 10^{-5}$	$8.2 \cdot 10^{-5}$	$1.8 \cdot 10^{-5}$	$2.0 \cdot 10^{-4}$	$1.7 \cdot 10^{-4}$	$1.1 \cdot 10^{-4}$
ET	$4.5 \cdot 10^{-3}$	$4.2 \cdot 10^{-5}$	$3.5 \cdot 10^{-5}$	$5.1 \cdot 10^{-5}$	$4.3 \cdot 10^{-5}$	$8.5 \cdot 10^{-5}$	$2.7 \cdot 10^{-5}$	$1.9 \cdot 10^{-4}$	$3.8 \cdot 10^{-4}$	$1.8 \cdot 10^{-4}$
ERT	$4.0 \cdot 10^{-3}$	$3.7 \cdot 10^{-5}$	$3.4 \cdot 10^{-5}$	$4.0 \cdot 10^{-5}$	$\mathbf{3.6 \cdot 10^{-5}}$	$5.7 \cdot 10^{-5}$	$1.6 \cdot 10^{-5}$	$\mathbf{1.3 \cdot 10^{-4}}$	$\mathbf{1.6 \cdot 10^{-4}}$	$\mathbf{4.2 \cdot 10^{-5}}$
GBM	$4.0 \cdot 10^{-3}$	$3.6 \cdot 10^{-5}$	$3.1 \cdot 10^{-5}$	$3.9 \cdot 10^{-5}$	$3.9 \cdot 10^{-5}$	$5.5 \cdot 10^{-5}$	$1.6 \cdot 10^{-5}$	$1.7 \cdot 10^{-4}$	$2.2 \cdot 10^{-4}$	$4.7 \cdot 10^{-5}$
HGBM	$3.4 \cdot 10^{-3}$	$3.6 \cdot 10^{-5}$	$\mathbf{2.5 \cdot 10^{-5}}$	$\mathbf{3.2 \cdot 10^{-5}}$	$3.8 \cdot 10^{-5}$	$\mathbf{4.5 \cdot 10^{-5}}$	$1.6 \cdot 10^{-5}$	$2.5 \cdot 10^{-4}$	$5.0 \cdot 10^{-4}$	$7.4 \cdot 10^{-5}$
XGBM	$3.5 \cdot 10^{-3}$	$7.1 \cdot 10^{-5}$	$1.0 \cdot 10^{-4}$	$1.3 \cdot 10^{-4}$	$8.7 \cdot 10^{-5}$	$1.5 \cdot 10^{-4}$	$6.0 \cdot 10^{-5}$	$5.4 \cdot 10^{-4}$	$7.5 \cdot 10^{-4}$	$2.1 \cdot 10^{-4}$
LGBM	$3.5 \cdot 10^{-3}$	$3.7 \cdot 10^{-5}$	$2.6 \cdot 10^{-5}$	$3.3 \cdot 10^{-5}$	$3.8 \cdot 10^{-5}$	$4.6 \cdot 10^{-5}$	$1.6 \cdot 10^{-5}$	$2.2 \cdot 10^{-4}$	$4.7 \cdot 10^{-4}$	$6.2 \cdot 10^{-5}$

and Root Mean Square Error (RMSE). The MAE is widely employed in forecasting literature [64] and is calculated to assess the accuracy of a model $\widetilde{A}_{\_,k}$ in estimating $A_{\_,k}$ as follows:

$$\mathrm{MAE}_{\widetilde{A}_{\_,k}} = \frac{1}{\left|\mathcal{D}_{\mathrm{test},A_{\_,k}}\right|} \sum_{(x^i,A^i_{\_,k})\in\mathcal{D}_{\mathrm{test},A_{\_,k}}} \left|A^i_{\_,k} - \widetilde{A}_{\_,k}(x^i)\right| \tag{4}$$

Additionally, a model $\widetilde{A}_{\_,k}$ is evaluated also in terms of RMSE, calculated using the following formula:

$$\mathrm{RMSE}_{\widetilde{A}_{\_,k}} = \sqrt{\frac{1}{\left|\mathcal{D}_{\mathrm{test},A_{\_,k}}\right|} \sum_{(x^i,A^i_{\_,k})\in\mathcal{D}_{\mathrm{test},A_{\_,k}}} \left(A^i_{\_,k} - \widetilde{A}_{\_,k}(x^i)\right)^2} \tag{5}$$

Analogously, $\mathrm{MAE}_{\widetilde{\phi}_{\_,k}}$ and $\mathrm{RMSE}_{\widetilde{\phi}_{\_,k}}$ are employed to evaluate the accuracy of a model $\widetilde{\phi}_{\_,k}$. MAE is a robust indicator of performance, providing insights into the average magnitude of errors without emphasizing the influence of outliers [65]. On the other hand, RMSE emphasizes the impact of outliers and tends to be more sensitive to extreme errors. Thus, the combination of MAE and RMSE enables a comprehensive evaluation of model accuracy, offering a balanced perspective on both typical and worst-case performance scenarios.

3.7. Forecasting results

Table 2 presents a comprehensive overview of the $\mathrm{MAE}_{\widetilde{A}_{\_,k}}$ values associated with the ML models presented in Section 3.4 and fitted according to the specifications outlined in Section 3.5. Each column of the table represents a distinct harmonic order k, with each row corresponding to a different ML model. The minimum value in each column is bolded to highlight the best-found predictor. Similarly, Table 3 provides an overview of the $\mathrm{RMSE}_{\widetilde{A}_{\_,k}}$ values. For each harmonic order k, the consistency of both statistical indicators in identifying the same best predictors underscores the reliability of the obtained results. For each harmonic order k, the selected best-predictor is reported in the second column of Table 6.

Regarding the selection of $\widetilde{\phi}^*_{\_,k}$, Tables 4 and 5 show the errors associated with predicting $\phi_{\_,k}$ measured in terms of MAE and RSME, respectively. In this case, various models demonstrate comparable performance levels. Table 6 details, for each harmonic order k, the chosen models that collectively demonstrate the minimum values for both $\mathrm{MAE}_{\widetilde{\phi}_{\_,k}}$ and $\mathrm{RMSE}_{\widetilde{\phi}_{\_,k}}$. The choice of the models for the application is finalized in Section 5 below. The findings demonstrate the efficacy of the implemented ML methods in accurately predicting TE. Consistently with the literature, classical ML models work effectively when dealing with limited training data [49,50]. Additionally, ELM was implemented for comparison with the literature [38,51], despite this model not being compatible with integration into the TwinCAT environment. ELM consistently outperformed MLP but exhibited less accurate predictions in terms of MAE and RMS compared with all the other methods implemented.

4. Implementation of ML-driven compensation

A novel PLC-based framework enabling ML-driven online motion compensation has been defined as shown in Fig. 5. The system reads as input the custom motion profile, $\theta_{out}(t)$, to be obtained at the reducer output shaft, stored into an external CSV file. The same file also provides the speed and resistant (i.e. externally applied) torque profiles, namely $\omega_{out} = \dot{\theta}_{out}$ and M_{out}. These vectors are then utilized within the two main PLC control modules, addressing respectively the ML model-based motion compensation on the MS and the torque control on the LS to achieve the correct loading on the tested RV reducer. As it can be seen from the schematic, both these modules also make use of real-time sensor data from the rig to properly compute the motion and torque setpoints to be sent to the servomotors drives during the experiment. In particular, the MS servomotor is position-controlled (having the position/speed/current loops managed into the drive), whereas the LS servomotor is torque-controlled (having only the internal current loop managed at drive level).

To integrate the ML models discussed in Section 3 and exported as ONNX files into TwinCAT (i.e. the automation software by Beckhoff), it is necessary to install the TF38xx package on both the PLC and the lab PC. This package facilitates the deployment of specialized Function Blocks (FB) for ML tasks and the handling of ONNX files. As highlighted in Section 3, the selection of ML models should align with those accessible within the library.

Table 4
$\mathrm{MAE}_{\tilde{\phi}_{\_,k}}$ values of the trained ML models.

Model	Harmonics order k								
	1	3	39	40	78	81	156	162	240
SVM	$2.2 \cdot 10^{-3}$	$3.3 \cdot 10^{-2}$	$2.7 \cdot 10^{-2}$	$6.1 \cdot 10^{-2}$	$1.9 \cdot 10^{-1}$	$1.3 \cdot 10^{-1}$	1.2	$4.9 \cdot 10^{-1}$	$4.9 \cdot 10^{-1}$
MLP	$7.2 \cdot 10^{-3}$	$6.5 \cdot 10^{-2}$	$6.2 \cdot 10^{-2}$	$8.0 \cdot 10^{-2}$	$1.6 \cdot 10^{-1}$	$1.5 \cdot 10^{-1}$	1.9	$7.8 \cdot 10^{-1}$	$7.0 \cdot 10^{-1}$
RF	$2.0 \cdot 10^{-3}$	$2.4 \cdot 10^{-2}$	$2.8 \cdot 10^{-2}$	$3.7 \cdot 10^{-2}$	**$7.4 \cdot 10^{-2}$**	**$5.3 \cdot 10^{-2}$**	**$5.1 \cdot 10^{-1}$**	$2.3 \cdot 10^{-1}$	$2.5 \cdot 10^{-1}$
DT	$2.1 \cdot 10^{-3}$	$3.0 \cdot 10^{-2}$	$3.6 \cdot 10^{-2}$	$4.3 \cdot 10^{-2}$	$9.0 \cdot 10^{-2}$	$6.6 \cdot 10^{-2}$	$5.2 \cdot 10^{-1}$	**$2.0 \cdot 10^{-1}$**	**$2.3 \cdot 10^{-1}$**
ET	$2.4 \cdot 10^{-3}$	$3.1 \cdot 10^{-2}$	$3.5 \cdot 10^{-2}$	$5.1 \cdot 10^{-2}$	$9.4 \cdot 10^{-2}$	$8.7 \cdot 10^{-2}$	$7.1 \cdot 10^{-1}$	$2.8 \cdot 10^{-1}$	$2.6 \cdot 10^{-1}$
ERT	$2.2 \cdot 10^{-3}$	$2.7 \cdot 10^{-2}$	$2.8 \cdot 10^{-2}$	$4.0 \cdot 10^{-2}$	$7.6 \cdot 10^{-2}$	$5.6 \cdot 10^{-2}$	$5.3 \cdot 10^{-1}$	**$2.0 \cdot 10^{-1}$**	**$2.3 \cdot 10^{-1}$**
GBM	$2.0 \cdot 10^{-3}$	$2.4 \cdot 10^{-2}$	$3.0 \cdot 10^{-2}$	**$3.6 \cdot 10^{-2}$**	**$7.4 \cdot 10^{-2}$**	**$5.3 \cdot 10^{-2}$**	$5.4 \cdot 10^{-1}$	$2.5 \cdot 10^{-1}$	$2.9 \cdot 10^{-1}$
HGBM	$1.9 \cdot 10^{-3}$	**$2.0 \cdot 10^{-2}$**	**$2.1 \cdot 10^{-2}$**	$4.0 \cdot 10^{-2}$	$9.1 \cdot 10^{-2}$	$5.7 \cdot 10^{-2}$	$7.4 \cdot 10^{-1}$	$3.5 \cdot 10^{-1}$	$3.6 \cdot 10^{-1}$
XGBM	$1.9 \cdot 10^{-3}$	$2.4 \cdot 10^{-2}$	$3.2 \cdot 10^{-2}$	$6.1 \cdot 10^{-2}$	$1.4 \cdot 10^{-1}$	$9.1 \cdot 10^{-2}$	$9.6 \cdot 10^{-1}$	$5.4 \cdot 10^{-1}$	$3.9 \cdot 10^{-1}$
LGBM	**$1.8 \cdot 10^{-3}$**	$2.1 \cdot 10^{-2}$	**$2.1 \cdot 10^{-2}$**	$4.0 \cdot 10^{-2}$	$9.5 \cdot 10^{-2}$	$5.5 \cdot 10^{-2}$	$7.4 \cdot 10^{-1}$	$3.5 \cdot 10^{-1}$	$3.4 \cdot 10^{-1}$

Table 5
$\mathrm{RMSE}_{\tilde{\phi}_{\_,k}}$ values of the trained ML models.

Model	Harmonics order k								
	1	3	39	40	78	81	156	162	240
SVM	$3.1 \cdot 10^{-3}$	$4.2 \cdot 10^{-2}$	$4.4 \cdot 10^{-2}$	$9.7 \cdot 10^{-2}$	$3.2 \cdot 10^{-1}$	$2.0 \cdot 10^{-1}$	1.8	1.1	1.1
MLP	$1.3 \cdot 10^{-2}$	$8.4 \cdot 10^{-2}$	$7.7 \cdot 10^{-2}$	$1.1 \cdot 10^{-1}$	$2.4 \cdot 10^{-1}$	$2.2 \cdot 10^{-1}$	2.2	1.2	1.1
RF	$2.8 \cdot 10^{-3}$	$3.3 \cdot 10^{-2}$	$4.3 \cdot 10^{-2}$	**$5.5 \cdot 10^{-2}$**	**$1.6 \cdot 10^{-1}$**	**$8.2 \cdot 10^{-2}$**	**1.2**	$6.8 \cdot 10^{-1}$	$6.3 \cdot 10^{-1}$
DT	$2.8 \cdot 10^{-3}$	$4.2 \cdot 10^{-2}$	$6.1 \cdot 10^{-2}$	$6.1 \cdot 10^{-2}$	$2.0 \cdot 10^{-1}$	$1.0 \cdot 10^{-1}$	1.3	$7.3 \cdot 10^{-1}$	$6.7 \cdot 10^{-1}$
ET	$3.3 \cdot 10^{-3}$	$4.6 \cdot 10^{-2}$	$6.2 \cdot 10^{-2}$	$7.4 \cdot 10^{-2}$	$2.3 \cdot 10^{-1}$	$1.5 \cdot 10^{-1}$	1.5	$9.3 \cdot 10^{-1}$	$6.8 \cdot 10^{-1}$
ERT	$3.6 \cdot 10^{-3}$	$4.0 \cdot 10^{-2}$	$4.4 \cdot 10^{-2}$	$6.0 \cdot 10^{-2}$	$1.8 \cdot 10^{-1}$	$1.1 \cdot 10^{-1}$	**1.2**	**$6.4 \cdot 10^{-1}$**	**$5.8 \cdot 10^{-1}$**
GBM	$2.6 \cdot 10^{-3}$	$3.4 \cdot 10^{-2}$	$4.5 \cdot 10^{-2}$	**$5.5 \cdot 10^{-2}$**	$1.8 \cdot 10^{-1}$	$8.4 \cdot 10^{-2}$	1.3	$7.1 \cdot 10^{-1}$	$7.1 \cdot 10^{-1}$
HGBM	**$2.5 \cdot 10^{-3}$**	**$2.9 \cdot 10^{-2}$**	**$2.7 \cdot 10^{-2}$**	$6.0 \cdot 10^{-2}$	$1.9 \cdot 10^{-1}$	$8.5 \cdot 10^{-2}$	1.3	$7.0 \cdot 10^{-1}$	$7.4 \cdot 10^{-1}$
XGBM	$2.8 \cdot 10^{-3}$	$3.3 \cdot 10^{-2}$	$4.3 \cdot 10^{-2}$	$8.9 \cdot 10^{-2}$	$2.3 \cdot 10^{-1}$	$1.3 \cdot 10^{-1}$	1.4	$8.1 \cdot 10^{-1}$	$7.6 \cdot 10^{-1}$
LGBM	**$2.5 \cdot 10^{-3}$**	$3.0 \cdot 10^{-2}$	$2.8 \cdot 10^{-2}$	$6.0 \cdot 10^{-2}$	$1.9 \cdot 10^{-1}$	**$8.2 \cdot 10^{-2}$**	1.3	$7.0 \cdot 10^{-1}$	$7.1 \cdot 10^{-1}$

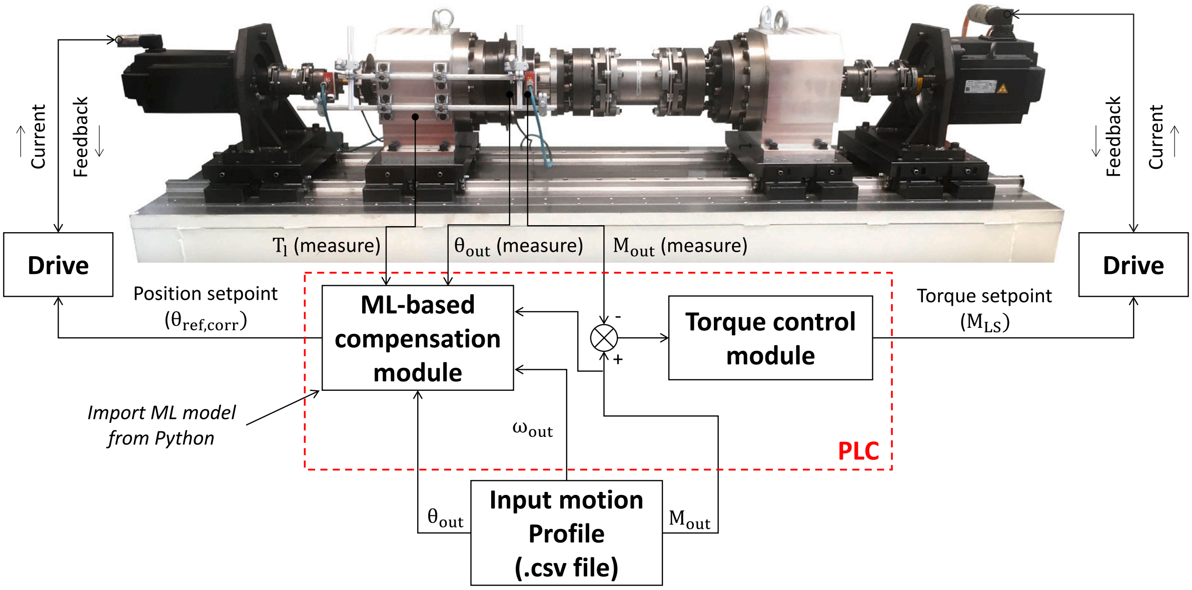
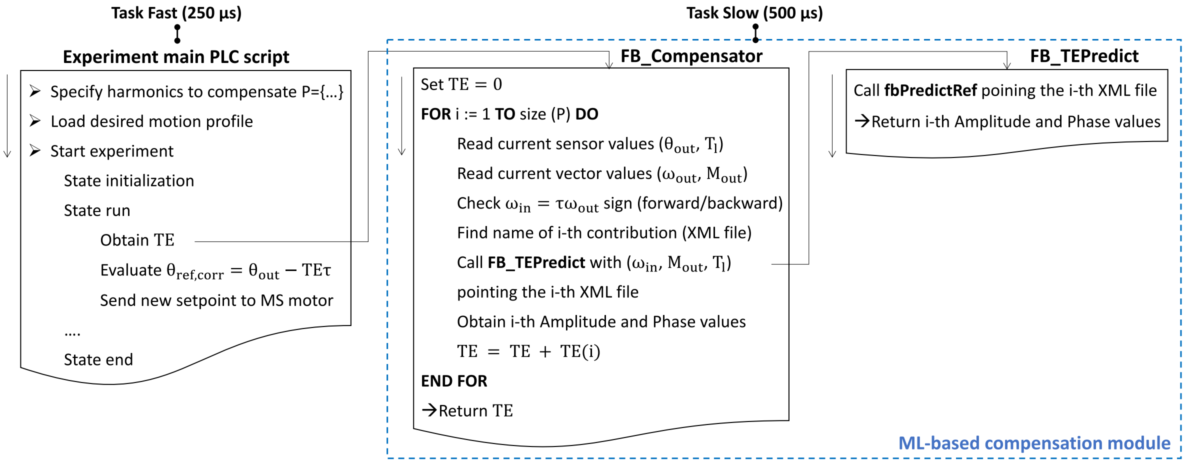


Fig. 5. Schematic of the novel test rig controller with ML-based motion compensation.

4.1. ML-based compensation module

The proposed compensation module comprises two FB, named *FB_TEPredict* and *FB_Compensator*, as illustrated in Fig. 6. These FB are programmed to cyclically extrapolate the TE value from the imported ML models and subsequently compute the compensation value at each PLC cycle. In particular, the *FB_TEPredict*, operating at lower level, considers as inputs the sensor data T_l and the reference data from the input motion profile ($\omega_{in} = \tau\omega_{out}$ and M_{out}), as visible in Fig. 5. It shall be noted that the ONNX files exported from Python (two for each harmonic $k \in P$, i.e. one for $A_{\_,k}$ and one for $\phi_{\_,k}$) cannot be readily processed by the PLC and are therefore converted into XML files within the TwinCAT environment. Subsequently, to simplify the ML files management, the amplitude-related and phase-related XML files are merged together to form two comprehensive files. This is valid under the condition that every original file is created selecting the same ML model in Python. Consequently, combining models of different types, such as a HGBM model and an SVM model for two spectral contributions, requires importing two separate XML files, one for HGBM and one for SVM.

At a higher level, the *FB_Compensator* employs a *FOR* loop within a single PLC cycle to gather the $A_{\_,k}$ and $\phi_{\_,k}$ values for each harmonic by invoking *FB_TEPredict* (see Fig. 6). Subsequently, it computes the current TE as in Eq. (3), with θ_{out} coming from the sensor and in this case expressed in radians. This information is then used in the main PLC script to determine the corrected position setpoint ($\theta_{ref,corr}$) for the MS



```
Experiment main PLC script
➢ Specify harmonics to compensate P={…}
➢ Load desired motion profile
➢ Start experiment
    State initialization
    State run
        Obtain TE
        Evaluate $\theta_{ref,corr} = \theta_{out} - TE\tau$
        Send new setpoint to MS motor
    ….
    State end
```

```
FB_Compensator
Set TE = 0
FOR i := 1 TO size (P) DO
    Read current sensor values ($\theta_{out}$, $T_l$)
    Read current vector values ($\omega_{out}$, $M_{out}$)
    Check $\omega_{in} = \tau\omega_{out}$ sign (forward/backward)
    Find name of i-th contribution (XML file)
    Call FB_TEPredict with ($\omega_{in}$, $M_{out}$, $T_l$)
    pointing the i-th XML file
    Obtain i-th Amplitude and Phase values
    TE = TE + TE(i)
END FOR
→Return TE
```

```
FB_TEPredict
Call fbPredictRef poining the i-th XML file
→Return i-th Amplitude and Phase values
```

Fig. 6. Logical structure of the ML-based compensation module.

Table 6
Top-performing models.

k	$\widetilde{A}^*_{\_,k}$	$\widetilde{\phi}^*_{\_,k}$
0	SVM	–
1	RF	LGBM
3	HGBM	HGBM
39	HGBM	HGBM
40	ERT	GBM
78	HGBM	RF
81	RF	RF
156	ERT	RF
162	ERT	ERT
240	ERT	ERT

servomotor as follows:

$$\begin{cases} \theta_{ref,corr} = \theta_{out} - TE\tau, & \omega_{in} \geq 100 \text{ rpm} \\ \theta_{ref,corr} = \theta_{out} - TE\frac{\tau\omega_{in}}{150}, & \omega_{in} < 100 \text{ rpm} \end{cases} \tag{6}$$

As evident from Eq. (6), at lower speeds the correction is weighted to mitigate potential instabilities and maintain proper rig functionality. Then, beyond the 100 rpm threshold, the compensation is fully enabled. It is important to highlight that while the main experiment script operates on a fast PLC task (cycle time of 250 μs), the two FB operate on a slower PLC task (cycle time of 500 μs) due to the higher computational load related to the use of ML models. An accurate signal synchronization is thus required, as described next.

4.2. Signal synchronization and control performance

The signals time synchronization is a key aspect to perform fine compensations in a distributed system. During the experiment, the acquisition and control operations are handled with a Beckhoff CX5140 PLC, which communicates with external devices via EtherCAT fieldbus, enabling high-performance and deterministic signal exchanges during operation. The input signals (sensor data, in particular θ_{out}, M_{out} and T_l) and output signals (position and torque setpoints for drive systems, i.e. $\theta_{ref,corr}$ and M_{LS}) are received and sent at a rate of 4000 Hz, being the main PLC and EtherCAT cycle time set equal to 250 μs. However, the implementation of the slower task (500 μs) executing the TE evaluation from the imported ML models and its communication with the main task introduced delays in data exchange. Variables are exchanged at the beginning of tasks, leading to longer overall times for the compensation. To determine the maximum achievable correction frequency of the proposed PLC-based motion compensation system, the time diagram reported in Fig. 7 (time values are accessible to the user in TwinCAT) must be carefully analyzed.

Table 7
Input parameters considered for ML prediction tests.

Test	ω_{in} [rpm]	M_{out} [N m]	T_l [°C]
1	350	650	26
2	775	1525	31
3	1650	370	27

The time interval between acquiring a new value (θ_{out}) from the SS encoder and applying the correction is approximately 1325.3 μs, setting the correction upper limit at 754.5 Hz. Now, considering that the highest TE peak in Fig. 3 is $f/f_{0,out} = 240$, the maximum speed at SS for correcting this contribution becomes $\omega_{out} = 188.6$ rpm. This value significantly exceeds the current imposed limit of $\omega_{in,max} = 1800$ rpm (as detailed in Section 2.1), resulting in $\omega_{out,max} = 22.2$ rpm. Given that the primary significant contributions occur at lower frequencies ($f/f_{0,out} \leq 100$ as illustrated in Fig. 3), the aforementioned limits are further relaxed in practical applications.

5. Experimental validation

This section details the experimental validation of the proposed online ML-based compensation, assessing its practical effectiveness in executing customized motion profiles using an industrial PLC. Before delving into the compensation process, the capabilities of the developed FB to predict the reducer TE within the considered operational domain are tested.

5.1. Preliminary ML model predictions test

To evaluate the prediction capabilities of the ML model in TwinCAT, three additional scenarios beyond the initial training dataset, comprising 1026 samples, have been chosen. The relevant parameters are outlined in Table 7. It shall be remarked that in these experiment a constant input parameter set (i.e., without altering ω_{in}, M_{out}, and T_l) is enforced throughout the experiment duration (0° $\leq \theta_{out} \leq$ 360°), as done during the model training. For each scenario, experiments are conducted on the rig to obtain sensor data. The experimental TE is then compared with the TE predicted by the FB, focusing on spectral components 0, 1, 39, and 40, identified as the most influential in Section 2.2. Regarding the ML models discussed in Section 3, with reference to Table 6, SVM is chosen for component 0, RF and LGBM are chosen for component 1, HGBM is chosen for component 39, whereas ERT and GBM are chosen for component 40. The comparison results are presented in Fig. 8, showing the forward TE curves. Generally, a good match between the curves is observed, with mean percentage errors of

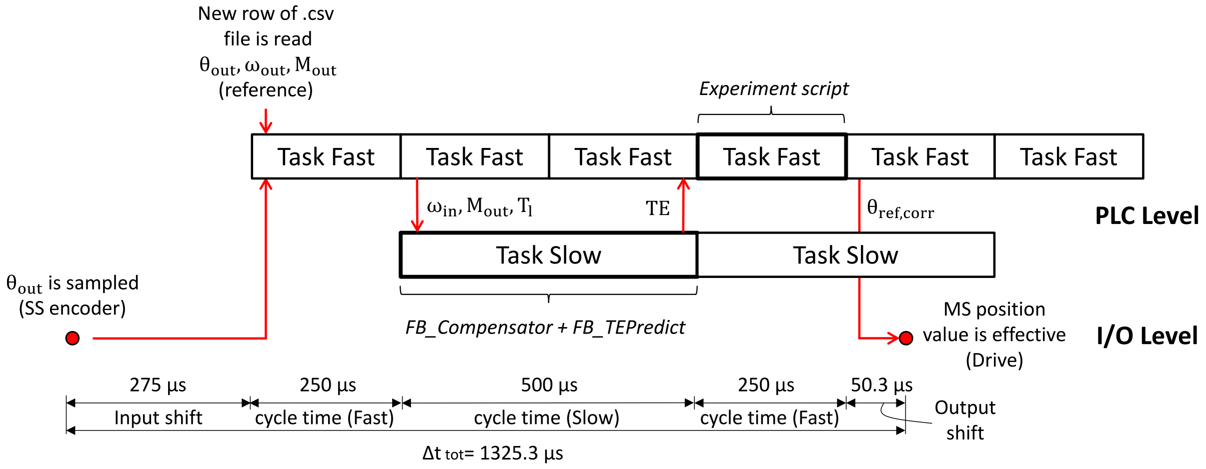


Fig. 7. Time diagram of the signals and data exchange within the PLC.

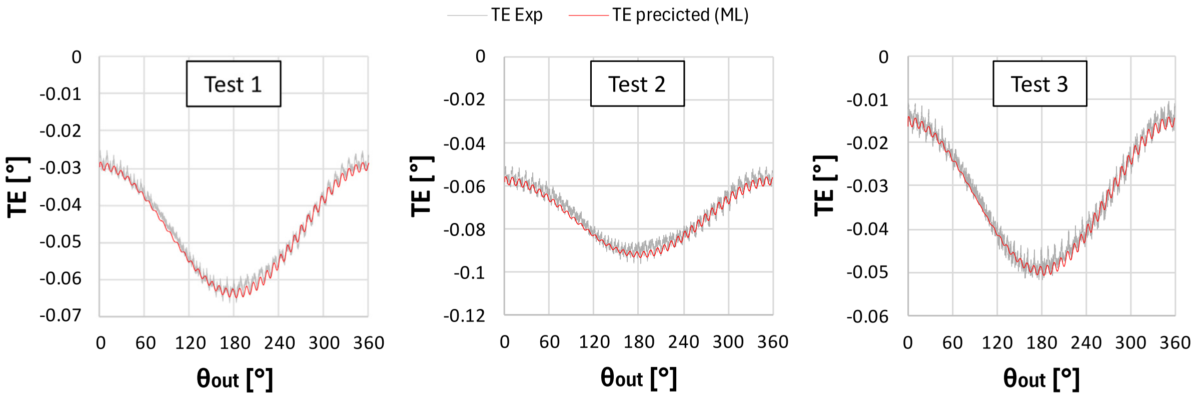


Fig. 8. Comparison between experimental and ML predicted TE functions in preliminary validation tests.

2.6%, 3.1% and 4.7% along the entire TE function for tests 1, 2, and 3, respectively. While incorporating additional harmonic contributions into the FB could potentially yield enhanced results, it is crucial to assess the associated increase in computational load. In practical terms, considering more than four contributions would necessitate extending the cycle time of the slow task (e.g., from 500 μs to 1 ms), thereby compromising control bandwidth performance, as detailed in Section 4.2 and illustrated in Fig. 7.

5.2. Motion profile compensation

Building upon the results shown in Section 5.1, here the ML capabilities are tested by imposing continuously variable input quantities, namely by applying time-varying motion $\theta_{out}(t)$ and resistant load $M_{out}(t)$ profiles as per Fig. 5. Two different motion profiles have been utilized during the experiment. The first profile is derived directly from observing the performance of a KUKA KR 210 R2700 Prime industrial robot (shown in Fig. 1) executing a pick-and-place operation. In this instance, the behavior of the robot's first joint is monitored and recorded in a CSV file, which is subsequently utilized on the test rig. For the second motion profile, a standard cycloidal law commonly encountered in industrial automation is employed, generated using the following relation:

$$\theta_{out}(t) = \theta_{out,0} + h\left(\frac{t-t_0}{T} - \frac{1}{2\pi}\sin\frac{2\pi(t-t_0)}{T}\right) \tag{7}$$

The motion law is defined with initial conditions $\theta_{out,0} = 0$, $t_0 = 0$, and by setting a time period $T = 20$ s and a final position $h = 360°$, enabling the spacing of the complete angular range of the output shaft over 20 s. A similar law is also used to deliver the dynamic load $M_{out}(t)$ to the reducer output shaft in the same time interval, reaching a maximum of 370 N m. The two motion profiles under consideration are depicted in Fig. 9, whereas their characteristics are detailed in Table 8. In line with Section 5.1, the selected harmonics for compensation include 0, 1 and 39. Two additional tests have been conducted to evaluate the impact of including components 40 and 78 across various operating conditions. Also in this case, the imported ML models utilize the algorithms reported in Table 6.

Table 8
Characteristics of imposed motion profiles.

Motion profile	$\omega_{in,max}$ [rpm]	$M_{out,max}$ [N m]	T_l [°C]
Robot	303	759	31.6
Cycloidal	500	370	26.7

The positive effect of enabling the ML-based compensation can be viewed in the plots reported in Fig. 10, where the original (uncompensated) TE is compared with the ones obtained when enabling compensation. In particular, the TE has been reduced of more than 80% and 90% for the robotic and cycloidal profile respectively, as also visible in Table 9, where both the Root Mean Square (RMS) and maximum values are reported for each case. The results depicted in Fig. 10 illustrate that the experiment conducted with the robotic law presents suboptimal compensation in both the initial and final segments. This arises from the adoption of a more aggressive speed law, as visible in Fig. 9, which results in premature crossing of the 100 rpm threshold. Finally, the inclusion of component 78 demonstrates superior performance with the robotic law compared to the cycloidal

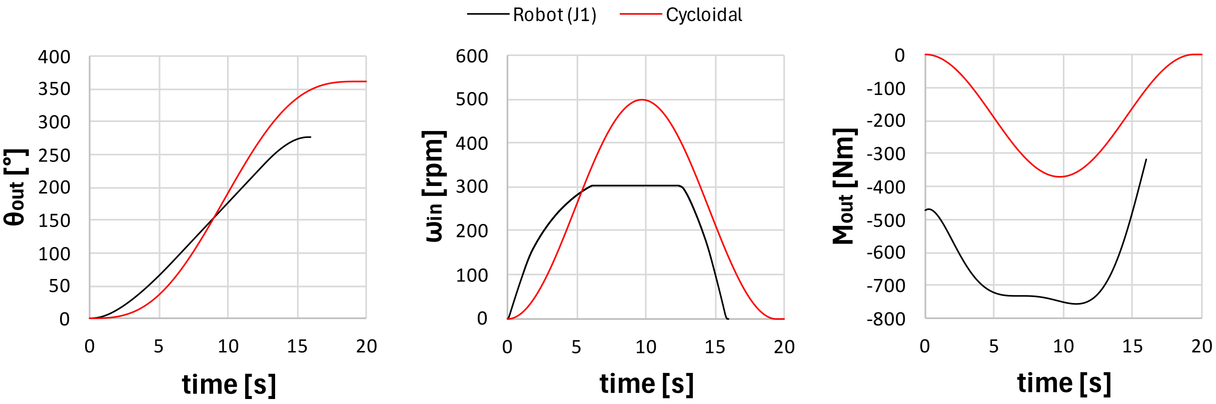


Fig. 9. Motion and load profiles adopted for validation.

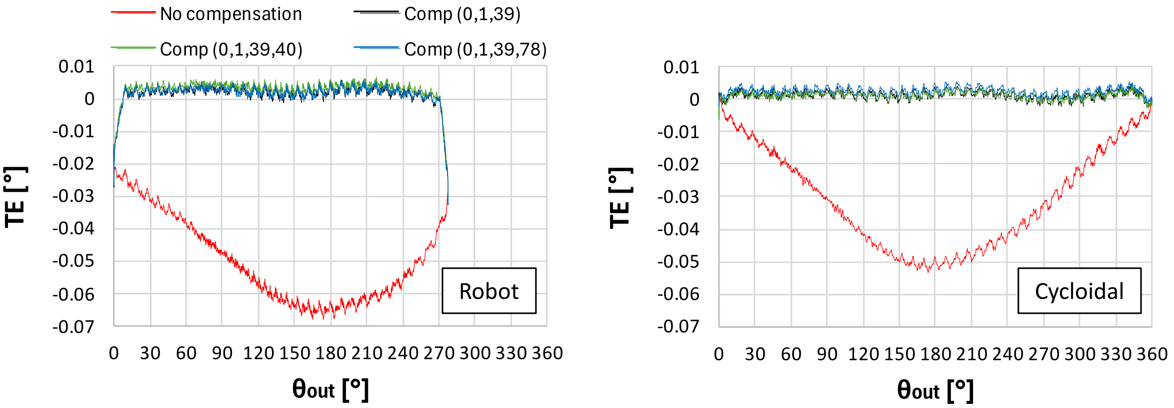


Fig. 10. Comparison between compensated and not compensated motion profiles.

one, in line with the preliminary analysis carried out in Section 2.2, which indicates its high dependency from M_{out}.

Overall, the experimental results demonstrate the validity of the proposed ML-based approach, showcasing accurate predictions and significant reductions in overall TE. The outcomes provide valuable insights into the real-world applicability and performance of the ML model in compensating for TE in industrial settings. In particular, the proposed method offers several practical and economic benefits for system integrators and equipment managers at the industrial level. First of all, the obtained improvements in position accuracy contribute to increase the operational efficiency and productivity in industrial settings, leading to potential gains in the production output, with higher product quality and reduced waste. Similar results can in fact be obtained on multi-axis systems such as serial industrial robots, where vendor-specific software interfaces allow to stream real-time correction data (e.g. up to 250 Hz, as in the compensation framework reported in Ref. [43] and based on the KUKA Robot Sensor Interface module). Also, the present approach eliminates the need for dedicated sensory equipment on real machines, which often pose installation challenges (see Ref. [66], where a secondary optical encoder has been installed on the robot to get joint-side position feedback) and significant extra costs. It relies on open-source libraries and standard exchange formats, facilitating seamless integration into existing commercial PLCs without the need for proprietary software modules. Moreover, substantial cost savings are achieved by streamlining development, tuning, and deployment phases of the motion compensator. The same approach can be extended to predictive maintenance, allowing to anticipate and address component wear or health issues proactively and hence preventing costly downtime and equipment failures.

Table 9
Obtained TE reductions (max values are expressed as absolute quantities).

Case	TE_{RMS} [°]	TE_{Max} [°]	Red. [%]
Robot			
No comp	0.0478	0.0681	–/–
Comp (0,1,39)	0.0080	0.0325	83.3/52.4
Comp (0,1,39,40)	0.0078	0.0309	83.6/54.7
Comp (0,1,39,78)	0.0079	0.0319	83.5/53.2
Cycloidal			
No comp	0.0282	0.0534	–/–
Comp (0,1,39)	0.0017	0.0044	94.0/91.7
Comp (0,1,39,40)	0.0017	0.0062	94.0/88.3
Comp (0,1,39,78)	0.0027	0.0020	90.5/96.3

6. Conclusions

This paper addresses the modeling and compensation of position errors within industrial servomechanisms. The study explores the use of ML algorithms to develop a comprehensive model of the TE, with a specific emphasis on RV reducers. Initially, a preliminary analysis is carried out to examine the spectral components of TE functions measured on a specialized test rig designed for evaluating the impact of input speed, applied torque, and oil temperature. Following this, an experimental campaign is carried out on the instrumented rig to effectively train several ML models capable of predicting the variation in the main TE peaks (amplitude and phase) based on the imposed operating conditions. The specific ML models, trained in Python, are selected based on their compatibility for being integrated into the Beckhoff PLC system. An evaluation test is conducted on all the trained ML models to assess and compare their predictive performance. In the subsequent phase of the research, the ML models are imported into TwinCAT to facilitate the implementation of online compensation strategies during the execution of custom motion profiles in high-dynamics servomechanisms. For this purpose, two FB are defined to cyclically extrapolate the new TE value from the imported ML models and subsequently calculate the corrected position to be streamed to the controlled servomotor. Experimental validation, carried out by applying both a motion law extracted from the first joint of a KUKA industrial robot during a pick-and-place operation and a standard cycloidal motion law, confirmed the effectiveness of the proposed approach.

The obtained results demonstrate precise predictions and substantial reductions over 90% in both the TE RMS and maximum values. In conclusion, this study highlights the potential of employing a data-driven approach for modeling and compensating position errors in industrial servomechanisms. The findings open avenues for improved accuracy and performance in automated production systems, showcasing the practical benefits of ML-driven approaches in addressing the challenges posed by complex and nonlinear components in industrial automation.

The gathered training dataset and the developed ML models are openly shared with the community, as an attempt to provide a strong basis for future advancements in compensation strategies for PLC-controlled servomechanisms. These could involve refining the proposed ML models or even developing more sophisticated ones (with a particular focus on reinforcement learning techniques) to predict the reducer wear or health issues and their related impact on the TE, also enabling more proactive maintenance interventions. Additionally, the integration into existing robot controllers could be beneficial, potentially increasing the accessibility of the proposed method to a wider range of industrial settings. Finally, conducting long-term studies would be essential to assess the effectiveness and sustainability of this method in real-world industrial environments for validating its long-term benefits and ensuring its practical feasibility.

Funding sources

This research was funded by the European Community's HORIZON 2020 programme under grant agreement No. 958303 (PeneloPe).

CRediT authorship contribution statement

Pietro Bilancia: Conceptualization, Methodology, Supervision, Validation, Writing – original draft, Writing – review & editing. **Alberto Locatelli:** Data curation, Investigation, Writing – original draft. **Alessio Tutarini:** Conceptualization, Data curation, Software. **Mirko Mucciarini:** Formal analysis, Software, Writing – original draft. **Manuel Iori:** Conceptualization, Supervision, Writing – review & editing. **Marcello Pellicciari:** Funding acquisition, Methodology, Writing – review & editing.

Declaration of competing interest

The authors 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 experimental data and trained ML models related to this work can be freely downloaded via a Mendeley Data link https://data.mendeley.com/datasets/g8mrry54j8/1.

References

[1] E. Oliva, G. Berselli, M. Pellicciari, A.O. Andrisano, An engineering method for the power flow assessment in servo-actuated automated machinery: Mechatronic modeling and experimental evaluation, Robot. Comput.-Integr. Manuf. 38 (2016) 31–41.

[2] A.-D. Pham, H.-J. Ahn, High precision reducers for industrial robots driving 4th industrial revolution: state of arts, analysis, design, performance evaluation and perspective, Int. J. Precis. Eng. Manuf.-Green Technol. 5 (2018) 519–533.

[3] A.-D. Pham, H.-J. Ahn, Rigid precision reducers for machining industrial robots, Int. J. Precis. Eng. Manuf. 22 (8) (2021) 1469–1486.

[4] Z. Qiu, J. Xue, Review of performance testing of high precision reducers for industrial robots, Measurement 183 (2021) 109794.

[5] S.-S. Yeh, H.-C. Su, Development of friction identification methods for feed drives of CNC machine tools, Int. J. Adv. Manuf. Technol. 52 (2011) 263–278.

[6] A. Hace, K. Jezernik, M. Terbuc, VSS motion control for a laser-cutting machine, Control Eng. Pract. 9 (1) (2001) 67–77.

[7] R.P. Borase, D. Maghade, S. Sondkar, S. Pawar, A review of PID control, tuning methods and applications, Int. J. Dyn. Control 9 (2021) 818–827.

[8] R. Fareh, S. Khadraoui, M.Y. Abdallah, M. Baziyad, M. Bettayeb, Active disturbance rejection control for robotic systems: A review, Mechatronics 80 (2021) 102671.

[9] H. Giberti, S. Cinquemani, G. Legnani, Effects of transmission mechanical characteristics on the choice of a motor-reducer, Mechatronics 20 (5) (2010) 604–610.

[10] P. Bilancia, L. Monari, R. Raffaeli, M. Peruzzini, M. Pellicciari, Accurate transmission performance evaluation of servo-mechanisms for robots, Robot. Comput.-Integr. Manuf. 78 (2022) 102400.

[11] L.X. Xu, J.L. Zhong, Y. Li, L. Chang, Design and dynamic transmission error analysis of a new type of cycloidal-pin reducer with a rotatable output-pin mechanism, Mech. Mach. Theory 181 (2023) 105218.

[12] M. Slamani, I.A. Bonev, Characterization and experimental evaluation of gear transmission errors in an industrial robot, Ind. Robot 40 (2013) 441–449.

[13] K.-Y. Wu, Y.-P. Shih, J.-J. Lee, Kinematic error analysis of the rotor vector gear reducer with machining tolerances, J. Braz. Soc. Mech. Sci. Eng. 42 (2020) 1–16.

[14] K.-S. Lin, K.-Y. Chan, J.-J. Lee, Kinematic error analysis and tolerance allocation of cycloidal gear reducers, Mech. Mach. Theory 124 (2018) 73–91, http://dx.doi.org/10.1016/j.mechmachtheory.2017.12.028, URL: https://www.sciencedirect.com/science/article/pii/S0094114X17312338.

[15] Nabtesco, Precision gearboxes - component sets, 2024, URL: https://www.nabtesco.de/en/products/precision-gearboxes-component-sets.

[16] S.D. Technologies, Zero backlash precision gearboxes, 2024, URL: https://us.sumitomodrive.com/en-us/motion-control/zero-backlash-precision-gearboxes.

[17] P.L. García, S. Crispel, E. Saerens, T. Verstraten, D. Lefeber, Compact gearboxes for modern robotics: A review, Front. Robot. AI 7 (2020) 103.

[18] R. Zhang, J. Zhou, Z. Wei, Study on transmission error and torsional stiffness of RV reducer under wear, J. Mech. Sci. Technol. 36 (8) (2022) 4067–4081.

[19] Y. Yang, G. Zhou, L. Chang, G. Chen, A modelling approach for kinematic equivalent mechanism and rotational transmission error of RV reducer, Mech. Mach. Theory 163 (2021) 104384.

[20] H. Xu, Z. Shi, B. Yu, H. Wang, Dynamic measurement of the lost motion of precision reducers in robots and the determination of optimal measurement speed, J. Adv. Mech. Des. Syst. Manuf. 13 (3) (2019).

[21] H. Wang, Z.-Y. Shi, B. Yu, H. Xu, Transmission performance analysis of RV reducers influenced by profile modification and load, Appl. Sci. 9 (19) (2019) 4099.

[22] L. Chen, H. Hu, Z. Zhang, X. Wang, Application of nonlinear output frequency response functions and deep learning to RV reducer fault diagnosis, IEEE Trans. Instrum. Meas. 70 (2020) 1–14.

[23] P. Peng, J. Wang, NOSCNN: A robust method for fault diagnosis of RV reducer, Measurement 138 (2019) 652–658.

[24] M. Cavazzuti, Optimization Methods: from Theory to Design Scientific and Technological Aspects in Mechanics, Springer Science & Business Media, 2012.

[25] M.A. Costa, B. Wullt, M. Norrlöf, S. Gunnarsson, Failure detection in robotic arms using statistical modeling, machine learning and hybrid gradient boosting, Measurement 146 (2019) 425–436.

[26] N.F. Espinoza Sepúlveda, J.K. Sinha, Blind application of developed smart vibration-based machine learning (SVML) model for machine faults diagnosis to different machine conditions, J. Vib. Eng. Technol. 9 (4) (2021) 587–596.

[27] P. Dev, S. Jain, P. Kumar Arora, H. Kumar, Machine learning and its impact on control systems: A review, Mater. Today Proc. 47 (2021) 3744–3749, 3rd International Conference on Computational and Experimental Methods in Mechanical Engineering.

[28] S. Thangavel, C. Maheswari, E. Priyanka, Dynamic modeling and control analysis of industrial electromechanical servo positioning system using machine learning technique, J. Test. Eval. 49 (4) (2021) 2425–2440.

[29] D. Kato, K. Yoshitsugu, T. Hirogaki, E. Aoyama, K. Takahashi, Predicting positioning error and finding features for large industrial robots based on deep learning, Int. J. Autom. Technol. 15 (2) (2021) 206–214.

[30] V. Bucinskas, A. Dzedzickis, M. Sumanas, E. Sutinys, S. Petkevicius, J. Butkiene, D. Virzonis, I. Morkvenaite-Vilkonciene, Improving industrial robot positioning accuracy to the microscale using machine learning method, Machines 10 (10) (2022) 940.

[31] P. Sethuramalingam, M. Uma, R. Garg, T. Pharlia, R. Rajsingh, Performance analysis of accuracy and repeatability of IRB1410 industrial robot using taguchi analysis with machine learning approach, Int. J. Interact. Des. Manufa. (IJIDeM) 17 (4) (2023) 1807–1821.

[32] E. Wescoat, M. Krugh, L. Mears, Random forest regression for predicting an anomalous condition on a UR10 cobot end-effector from purposeful failure data, Procedia Manuf. 53 (2021) 644–655.

[33] S. Yin, W. Ji, L. Wang, A machine learning based energy efficient trajectory planning approach for industrial robots, Procedia CIRP 81 (2019) 429–434, 52nd CIRP Conference on Manufacturing Systems (CMS), Ljubljana, Slovenia, June 12-14, 2019.

[34] J. Yakun, C. Jihong, Z. Huicheng, W. Junxiang, Contour error modeling and compensation of CNC machining based on deep learning and reinforcement learning, Int. J. Adv. Manuf. Technol. 2022/01/01 (2022).

[35] Pedregosa, et al., Scikit-learn: Machine learning in python, 2010, URL: https://scikit-learn.org/stable/index.html.

[36] W. Wang, W. Tian, W. Liao, B. Li, J. Hu, Error compensation of industrial robot based on deep belief network and error similarity, Robot. Comput.-Integr. Manuf. 73 (2022) 102220.

[37] Y. Bai, On the comparison of model-based and modeless robotic calibration based on a fuzzy interpolation method, Int. J. Adv. Manuf. Technol. 31 (11–12) (2007) 1243–1250.

[38] S. Ma, K. Deng, Y. Lu, X. Xu, Robot error compensation based on incremental extreme learning machines and an improved sparrow search algorithm, Int. J. Adv. Manuf. Technol. 125 (11–12) (2023) 5431–5443.

[39] S. Ferrarini, P. Bilancia, R. Raffaeli, M. Peruzzini, M. Pellicciari, A method for the assessment and compensation of positioning errors in industrial robots, Robot. Comput.-Integr. Manuf. 85 (2024) 102622.

[40] J. Li, L. Zou, G. Luo, W. Wang, C. Lv, Enhancement and evaluation in path accuracy of industrial robot for complex surface grinding, Robot. Comput.-Integr. Manuf. 81 (2023) 102521.

[41] M. Shahin, F.F. Chen, H. Bouzary, K. Krishnaiyer, Integration of lean practices and industry 4.0 technologies: smart manufacturing for next-generation enterprises, Int. J. Adv. Manuf. Technol. 107 (2020) 2927–2936.

[42] M. Belloni, P. Bilancia, R. Raffaeli, M. Peruzzini, M. Pellicciari, Design of a test rig for tuning and optimization of high dynamics servo-mechanisms employed in manufacturing automation, Procedia Manuf. 55 (2021) 48–55, FAIM 2021.

[43] S. Ferrarini, P. Bilancia, R. Raffaeli, M. Peruzzini, M. Pellicciari, A method for the assessment and compensation of positioning errors in industrial robots, Robot. Comput.-Integr. Manuf. 85 (2024) 102622.

[44] S. Jin, S. Shang, S. Jiang, M. Cao, Y. Wang, Sensitivity analysis of RV reducer rotation error based on deep Gaussian processes, Sensors 23 (7) (2023).

[45] M. Yamamoto, M. Iwasaki, H. Hirai, Y. Okitsu, K. Sasaki, T. Yajima, Modeling and compensation for angular transmission error in harmonic drive gearings, IEEJ Trans. Electr. Electron. Eng. 4 (2) (2009) 158–165.

[46] H.-N. Nguyen, J. Zhou, H.-J. Kang, A calibration method for enhancing robot accuracy through integration of an extended Kalman filter algorithm and an artificial neural network, Neurocomputing 151 (P3) (2015) 996–1005.

[47] H.-N. Nguyen, P.-N. Le, H.-J. Kang, A new calibration method for enhancing robot position accuracy by combining a robot model–based identification approach and an artificial neural network–based error compensation technique, Adv. Mech. Eng. 11 (1) (2019) 1687814018822935.

[48] H.Q. Cao, H.X. Nguyen, T.N.-C. Tran, H.N. Tran, J.W. Jeon, A robot calibration method using a neural network based on a butterfly and flower pollination algorithm, IEEE Trans. Ind. Electron. 69 (4) (2022) 3865–3875.

[49] M. Bai, M. Zhang, H. Zhang, M. Li, J. Zhao, Z. Chen, Calibration method based on models and least-squares support vector regression enhancing robot position accuracy, IEEE Access 9 (2021) 136060–136070.

[50] S. Ma, K. Deng, Y. Lu, X. Xu, Error compensation method of industrial robots considering non-kinematic and weak rigid base errors, Precis. Eng. 82 (2023) 304–315.

[51] G. Chen, J. Yang, H. Xiang, D. Ou, New positional accuracy calibration method for an autonomous robotic inspection system, J. Braz. Soc. Mech. Sci. Eng. 44 (5) (2022).

[52] A.J. Smola, B. Schölkopf, A tutorial on support vector regression, Stat. Comput. 14 (3) (2004) 199–222.

[53] Y. Lecun, Y. Bengio, G. Hinton, Deep learning, Nature 521 (7553) (2015) 436–444.

[54] J. Quinlan, Induction of decision trees, Mach. Learn. 1 (1) (1986) 81–106.

[55] P. Geurts, D. Ernst, L. Wehenkel, Extremely randomized trees, Mach. Learn. 63 (1) (2006) 3–42.

[56] L. Breiman, Random forests, Mach. Learn. 45 (1) (2001) 5–32.

[57] A. Natekin, A. Knoll, Gradient boosting machines, a tutorial, Front. Neurorobot. 7 (DEC) (2013).

[58] C. Bentéjac, A. Csörgő, G. Martínez-Muñoz, A comparative analysis of gradient boosting algorithms, Artif. Intell. Rev. 54 (3) (2021) 1937–1967.

[59] G. Ke, Q. Meng, T. Finley, T. Wang, W. Chen, W. Ma, Q. Ye, T.-Y. Liu, LightGBM: A highly efficient gradient boosting decision tree, Adv. Neural Inf. Process. Syst. 2017-December (2017) 3147–3155.

[60] T. Chen, C. Guestrin, XGBoost: A scalable tree boosting system, in: Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Vol. 13-17-August-2016, 2016, pp. 785–794.

[61] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, E. Duchesnay, Scikit-learn: Machine learning in python, J. Mach. Learn. Res. 12 (2011) 2825–2830.

[62] S. Arlot, A. Celisse, A survey of cross-validation procedures for model selection, Stat. Surv. 4 (2010) 40–79.

[63] X. He, K. Zhao, X. Chu, AutoML: A survey of the state-of-the-art, Knowl.-Based Syst. 212 (2021).

[64] F. Petropoulos, D. Apiletti, V. Assimakopoulos, M.Z. Babai, D.K. Barrow, S. Ben Taieb, C. Bergmeir, R.J. Bessa, J. Bijak, J.E. Boylan, J. Browell, C. Carnevale, J.L. Castle, P. Cirillo, M.P. Clements, C. Cordeiro, F.L. Cyrino Oliveira, S. De Baets, A. Dokumentov, J. Ellison, P. Fiszeder, P.H. Franses, D.T. Frazier, M. Gilliland, M.S. Gönül, P. Goodwin, L. Grossi, Y. Grushka-Cockayne, M. Guidolin, M. Guidolin, U. Gunter, X. Guo, R. Guseo, N. Harvey, D.F. Hendry, R. Hollyman, T. Januschowski, J. Jeon, V.R.R. Jose, Y. Kang, A.B. Koehler, S. Kolassa, N. Kourentzes, S. Leva, F. Li, K. Litsiou, S. Makridakis, G.M. Martin, A.B. Martinez, S. Meeran, T. Modis, K. Nikolopoulos, D. Önkal, A. Paccagnini, A. Panagiotelis, I. Panapakidis, J.M. Pavía, M. Pedio, D.J. Pedregal, P. Pinson, P. Ramos, D.E. Rapach, J.J. Reade, B. Rostami-Tabar, M. Rubaszek, G. Sermpinis, H.L. Shang, E. Spiliotis, A.A. Syntetos, P.D. Talagala, T.S. Talagala, L. Tashman, D. Thomakos, T. Thorarinsdottir, E. Todini, J.R. Trapero Arenas, X. Wang, R.L. Winkler, A. Yusupova, F. Ziel, Forecasting: theory and practice, Int. J. Forecast. 38 (3) (2022) 705–871.

[65] C.J. Willmott, K. Matsuura, Advantages of the mean absolute error (MAE) over the root mean square error (RMSE) in assessing average model performance, Clim. Res. 30 (1) (2005) 79–82.

[66] P. Mesmer, M. Neubauer, A. Lechler, A. Verl, Robust design of independent joint control of industrial robots with secondary encoders, Robot. Comput.-Integr. Manuf. 73 (2022) 102232.