Sensor Calibration using a Directional Trained Deep Neural Network

By W. Kolff

Master's Thesis
Robotics, Delft University of Technology

December 2024

Supervisors:
Prof. J. Kober
Prof. D. Dodou
Dr. T.C.T van Riet
drs. M. Beuling
Delft University of Technology & AMC

ACKNOWLEDGMENTS

I would like to thank Dimitra and Jens from TU Delft for their patience and persistence in supporting me throughout my master's thesis. And their sincerely sympathy for certain situations.

I would like to thank Dimitra and Dr. Jens Kober from the TU Delft for their support and valuable insights during this Thesis.

I am also very grateful for the opportunity to work with Tom, whose involvement was a great factor in maintaining my enthusiasm for the research. A special thanks goes to Maaike, who provided invaluable support in reviewing and assessing my work.

It was a pleasure to work with all of you, thanks a lot.

Sensor Calibration using a Directional Trained Deep Neural Network

By W. Kolff

***Abstract*— Force/torque sensors are essential tools that enable robots to effectively interact with their environments. Existing calibration methods often fail to capture inter-axis nonlinearities and coupling effects, particularly when available calibration data are sparse and discrete. To address this challenge, the presented approach employs a Deep Neural Network (DNN) that learns both the scaling and the direction of the input-output relationship. The method works by extracting the absolute magnitude and unit vector from the raw N-dimensional sensor output values, which can vary among sensors. The DNN takes this N-dimensional input and produces a 7-dimensional output—comprising a corrected 6D unit vector representing the desired force-torque direction and a scaling factor. The final measurement is then constructed by combining the output unit vector, the learned scaling factor, and the original input magnitude. This approach simplifies the calibration problem to a linear mapping along one axis, enabling the model to generalize well under limited training conditions while leveraging the DNN's strength in capturing nonlinear inter-axis relationships.**

The proposed DNN was trained and evaluated on both artificially generated and real-world datasets, and its performance was compared to two baseline models: a commonly used linear transformation model and a comparative DNN approach from the literature. On generated data, the proposed DNN achieved an RMSE of 36.9 ± 3.44, outperforming the comparative DNN (48.3 ± 4.47) and the linear transformation model (62.3 ± 0.76). Similar improvements were observed on the real-world dataset. Although these results are promising, they are based on artificially generated data and a single real-world dataset from one specific sensor. Further validation and more extensive testing are necessary. Nonetheless, the gains indicated here suggest meaningful potential for improved calibration strategies in force-controlled robotic applications, even under limited training conditions.

I. Introduction

With the advancement of more autonomous robotic systems across an increasing number of fields, such as surgery, the need to accurately sense and adapt to dynamic, unpredictable environments is also rising. For example, handling sensitive human tissue requires precise and reliable measurements of forces and torques (F/T) [1], [2], [3]. The precision of these measurements ensures safe and gentle manipulation of soft, variable materials. To achieve precise measurements, the sensors need to be accurately calibrated. Effective calibration translates raw electrical signals into meaningful data.

Current calibration methods face significant challenges. Traditional linear transformation matrices (LTMs) [4], [5], [6], [7], [8] often assume that sensor outputs are linear and decoupled. In practice, sensors can be non-linear and exhibit coupling effects, making these assumptions unreliable [9], [10], [11]. Recent data-driven approaches, including neural networks and hybrid models, address some of these issues but typically require large amounts of training data [12]. Acquiring these data can be expensive, time-consuming, and impractical [13], [14]. As a result, researchers and practitioners often choose between less accurate but data-efficient methods or more accurate methods that require extensive datasets and complex calibration procedures.

This work aims to develop a calibration method that balances limited data availability with the need for high accuracy. The main question is how to efficiently decouple and correct sensor outputs with minimal training data, while still handling nonlinearities and coupling effects.

A well-designed calibration method should be able to perform reliably even when only minimal training data is available. In an ideal situation it combines the data efficiency of a linear transformation model with the adaptability of a non-parametric approach. It handles minimal training data while capturing inter-axis nonlinearities and coupling effects. By combining these strengths, such a method would also generalize well beyond the training set, ensuring accurate force–torque measurements in diverse real-world scenarios.

Contributions

- DNN-Based Calibration Model: We introduce a data-efficient DNN approach that emphasizes direction and scaling, improving performance with limited datasets.
- Nonlinear and Multiaxial Correction: The model effectively corrects sensor outputs under multi-axial, interaxis nonlinear conditions, addressing key limitations of LTMs.
- Comparative analysis: We compare our approach with traditional LTM and existing DNN solutions, demonstrating improvements in accuracy and robustness.

We begin with a review of related work, describing existing calibration approaches and their limitations (see Chapter II). Next, we detail our methodology, including the model structure and training procedures (see Chapter III). Then, the experimental results are presented (see Chapter V). Finally, we summarize our findings and highlight areas for further research (see Chapter VI and Chapter VII).

II. Background & Related Work

As described in the introduction, force-torque sensors are crucial components in advanced robotic systems and are used in a wide range of applications. However, their accuracy is compromised by various errors inherent in the sensing

elements and structural design. Understanding these errors is important for effective calibration.

A. Sensor Errors and Their Impact on Calibration

Several types of errors affect force-torque sensors:

a) Nonlinearity: Inter-axis nonlinearity frequently arises from *structural* factors, such as small-angle approximations and simplified models used to describe the elastic components of force-torque sensors. These simplifications become inadequate when deformations are significant, particularly under large loads, leading to coupling between different axes [8]. In contrast, *intra-axis* nonlinearity typically stems from *material* factors, since the materials used in sensor construction often exhibit nonlinear stress-strain relationships [15]. Variations in material properties—due to manufacturing inconsistencies or environmental conditions can further worsen these effects. Both structural and material nonlinearities undermine the effectiveness of linear calibration methods

b) Creep: or drift refers to the gradual change in sensor output under a constant load over time [16]. This phenomenon is particularly pronounced in certain materials and can lead to drift in measurements. Creep introduces time-dependent errors that are not accounted for in static calibration models, necessitating frequent recalibration or the development of models that can compensate for temporal changes.

c) Noise: encompasses random fluctuations in the sensor signal caused by electrical interference, thermal variations, and other external factors [14]. Noise reduces the precision of measurements and can obscure the true signal, making it difficult to achieve high calibration accuracy without adequate filtering or modeling techniques.

d) Coupling: or crosstalk between sensor axes, occurs when applying force or torque along one axis does not affect only the corresponding sensing elements, but also unintentionally influences measurements on other axes [17]. This can happen for a variety of reasons. On a mechanical level, slight structural deformations may transfer force from one axis to another. Electrically, interference within the sensing circuits may cause readings to bleed into neighboring channels. Additionally, subtle differences among supposedly identical sensing elements, such as strain gauges, can lead to asymmetric responses. For instance, a positive force along the x-axis might predominantly stress one set of strain gauges, while a negative force along the same axis might stress a different set. If these gauges vary slightly in their properties due to manufacturing imperfections, the sensor's outputs may differ even for forces of the same magnitude but opposite direction, ultimately violating the assumption of axis independence in many calibration models [8].

B. Traditional Calibration Methods

Linear calibration methods using Linear Transformation Matrices (LTMs) are widely adopted due to their simplicity and computational efficiency [9], [18], [19]. LTMs map raw sensor outputs to force and torque values through an affine transformation:

$$\mathbf{W} = \mathbf{C}\mathbf{S} + \mathbf{b} \tag{1}$$

where:

- $\mathbf{W} = [F_x, F_y, F_z, \tau_x, \tau_y, \tau_z]^T$ is the wrench vector, representing the three force components (F_x, F_y, F_z) and three torque components (τ_x, τ_y, τ_z) in a Cartesian coordinate system.
- $\mathbf{S}$ is the sensor output vector, typically consisting of raw measurements (e.g., voltage or digital counts) from the sensor's sensing elements.
- $\mathbf{C}$ is the calibration matrix, a $6 \times n$ matrix (where n is the number of sensing elements) that linearly transforms the sensor outputs $\mathbf{S}$ into the wrench vector $\mathbf{W}$. Each element of $\mathbf{C}$ represents the sensitivity of a specific force or torque component to the corresponding sensor output.
- $\mathbf{b}$ is the offset vector, a 6×1 vector that accounts for biases or zero offsets in the sensor outputs. It ensures that the wrench vector $\mathbf{W}$ is zero when no external forces or torques are applied.

The calibration process involves determining $\mathbf{C}$ and $\mathbf{b}$ by applying known forces and torques (reference wrenches) to the sensor and recording the corresponding sensor outputs. The calibration parameters are optimized by minimizing the squared error between the reference wrenches ($\mathbf{W}_d$) and the measured wrenches ($\mathbf{W}_m$):

$$\text{Minimize Error}^2 = \sum_{i=1}^{N} \|\mathbf{W}_{d,i} - \mathbf{W}_{m,i}\|_2^2 \tag{2}$$

Here, N is the number of calibration samples, and the minimization ensures that the calibration matrix $\mathbf{C}$ and offset vector $\mathbf{b}$ provide the best fit for all observed data points. This process is achieved through the *least squares method (LSM)*, which solves the optimization problem by finding the parameters that minimize the residual error between the predicted and reference wrenches.

However, LTMs inherently assume that sensor outputs are both linear and decoupled, meaning that forces and torques along one axis do not influence the sensor outputs along other axes. In real-world applications, these assumptions are often violated due to coupling effects, where outputs along one axis are influenced by forces and torques applied along different axes [20], [21], [22]. These coupling effects can introduce significant errors, as discussed in Section II-A.

Researchers have proposed decoupled sensor designs using advanced structural arrangements and finite element analysis (FEA) [4], [6], [23], but such designs often remain susceptible to inter-axis interference when multiple forces are applied simultaneously.

C. Hybrid Calibration Methods

To enhance calibration accuracy, hybrid methods combine linear models with machine learning techniques to capture

residual nonlinearities. Ma et al. [15] proposed a hybrid approach that first applies an LTM to address the primary linear relationship, followed by a Support Vector Machine (SVM) to model the nonlinear residuals. The SVM captures discrepancies between the LTM's predictions and the actual measurements.

Simiarly Al-mai et al. [16] proposed a static nonlinear model (SSGPR). He utilized a linear calibration matrix to capture primary linear relationships and a static model to predict and correct nonlinear residuals between measured values and calibration matrix predictions.

Hybrid methods like those employing SVM or static nonlinear models demonstrate improvements in capturing nonlinear behaviors. However, they are effective primarily when the foundational error can be largely corrected by a linear transformation. They may struggle with more complex coupling errors.

D. Neural Network and Deep Learning Approaches

Advancements in machine learning have led to the exploration of neural networks (NNs) and deep neural networks (DNNs) for sensor calibration. Lu et al. [24] proposed a basic NN calibration framework but emphasized the need for diversified and uniformly sampled training data to prevent overfitting. Similarly, Wang et al. [14] explored NNs with limited depth, which restricted their capacity to model complex nonlinear interactions.

Cao et al. [21] introduced a two-step neural network (NN)-based calibration approach where the first NN corrects the primary errors, and the second NN compensates for interference forces. However, the second step uses linear activation functions, limiting its expressiveness. This limitation suggests that the entire approach could potentially be simplified into a single shallow NN.

Ming et al. [20] employed a Radial Basis Function (RBF) NN to mitigate coupling effects, offering reduced computational complexity compared to standard NNs. While promising, this method remained prone to overfitting and relied heavily on empirically tuned parameters.

Oh et al. [25] employed a Deep Neural Network (DNN) to address nonlinearities and coupling effects by training on data collected from a robotic arm applying multi-axis forces and torques, resulting in notable error reductions compared to more traditional methods. However, it remains unclear how well this model would generalize to a broader operational range, as its training data was limited to a single load configuration.

Osburg et al. [26] highlighted that robots are typically used within a limited operational range due to specialized applications. They showed that training a DNN within this specific range could achieve superior calibrations compaired to LTMs, but at the cost of broader applicability. Their results underscored that DNNs excel in specific, dense data domains but struggle to generalize to data outside of this limited train data range.

Despite their strengths, DNNs trained on direct input-output mappings struggle with extrapolation when the training data covers only a limited portion of the operational range. Large datasets are required to ensure effective learning.

E. Summarizing the Existing Calibration Methods

Existing calibration methods, both traditional and advanced, exhibit limitations. Linear models like LTMs fail to capture complex nonlinear behaviors and coupling effects, especially asymmetric and nonlinear scaling between different directions [21], [20]. Hybrid models improve upon linear methods but depend on extensive training data and may not effectively decouple intra-axis erros. DNNs trained directly on input-output mappings offer increased modeling capacity but require substantial training data [12], The issue is that most current calibration data collection techniques do not provide a continues data set covering the entire operational range. The current cheaper techniques create datasets that only cover specific regions of the operational range. This is because data is often collected by hanging weights [27], [22] or directly connecting weights [28], [24], [25], [9] to the surface of the sensors, which introduce sensor signals in fixed increments. This resulting in limited coverage of the full operational range. There are machines that can create continuous representative datasets but these can be complex and expensive [13].

There is a clear need for a calibration model that can effectively decouple the coupling effects between axes in force-torque sensors, without relying on exhaustive datasets. Such a model should generalize across the operational range to perform accurately with limited or discrete training data applicable to continuous real-world scenarios. Additionally, it should enhance practicality by reducing the need for extensive data collection and simplifying the calibration process.

III. Method

Building on insights from previous calibration methods, we propose a new approach aimed at learning both direction and scaling factors between input and output vectors. We separate a vector's length (magnitude) from its orientation (direction) and work with normalized inputs and outputs. This approach may help the model generalize to various force and torque levels, even when training data is limited. We use a deep neural network (DNN) to capture potential nonlinear interactions among axes, while still assuming that each axis has a linear relation when looking at its magnitude alone.

The goal of this section is to understand what the model aims to learn and how it is designed. We first explain the model training process, including pre-processing steps, learning objective, loss function, and provide the training details. We then outline the prediction process, followed by zooming in on the models architecture. Next, we present the models used for comparison. These are two baseline models the comparative DNN Model and the Linear Transformation Model (LTM). Practical aspects, such as data collection and performance evaluation, will be given in the Experimental Setup chapter IV.

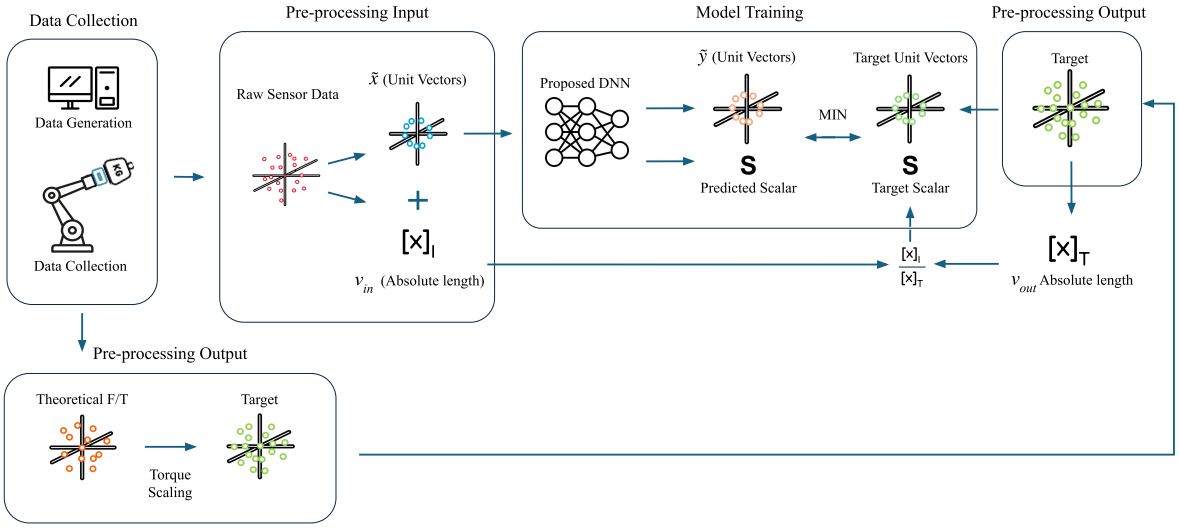


Fig. 1: The training process of the proposed DNN. This process consists of a data collection phase using both real and artificial data, multiple preprocessing steps, and the optimization of the proposed DNN.

A. Model Training process

Figure 1 illustrates the full pipeline of the training process, which learns two separate components from raw sensor data: a *direction* vector $\tilde{y}$, and a *scaling factor* s. Now follows the explanation of the training flow.

a) Pre-Processing: Data are generated by applying known forces and torques to the sensor setup or by simulation (see Chapter IV). The sensor provides raw measurements (e.g., strain-gauge voltage values). For instance, consider a six-dimensional voltage reading $\mathbf{x} \in R^6$ with components $(c_1, c_2, c_3, c_4, c_5, c_6)$. These readings are then transformed into $\tilde{\mathbf{x}}$ (unit vectors) and v_{in}, the absolute magnitude of the input signal:

$$v_{\text{in}} = \|\mathbf{x}\| = \sqrt{c_1^2 + c_2^2 + c_3^2 + c_4^2 + c_5^2 + c_6^2}, \tag{3}$$

$$\tilde{\mathbf{x}} = \frac{\mathbf{x}}{v_{\text{in}}}. \tag{4}$$

In parallel, theoretical force-torque (F/T) values or reference measurements undergo torque scaling to place torques on a comparable level with forces. Let $\mathbf{y} \in R^6$ represent the force-torque measurement with components $(f_1, f_2, f_3, \tau_1, \tau_2, \tau_3)$. To place forces and torques on comparable scales, compute

$$\alpha = \frac{F_{\max}}{\tau_{\max}}, \tag{5}$$

and multiply each torque component by α. Thus,

$$\mathbf{y} = \big(f_1,\ f_2,\ f_3,\ \alpha\tau_1,\ \alpha\tau_2,\ \alpha\tau_3\big). \tag{6}$$

Next, we obtain the target unit vectors and their corresponding absolute magnitudes, v_{out}. Specifically,

$$v_{\text{out}} = \|\mathbf{y}\| = \sqrt{f_1^2 + f_2^2 + f_3^2 + (\alpha\tau_1)^2 + (\alpha\tau_2)^2 + (\alpha\tau_3)^2}, \tag{7}$$

$$\text{Target Unit Vector} = \frac{\mathbf{y}}{v_{\text{out}}}. \tag{8}$$

By dividing v_{in} by v_{out}, we compute the target scaling factor. The combination of normalized inputs, scaled unit targets, and this target scaling factor forms the basis for training.

b) Direction and Scaling: The proposed deep neural network (DNN) learns to predict two key outputs from the normalized sensor input $\tilde{\mathbf{x}}$:

1) $\tilde{\mathbf{y}}$, the direction of the force-torque vector,
2) s, a scalar that captures the ratio of output magnitude to input magnitude.

By explicitly separating direction from magnitude, the model focuses on both "where" the force-torque vector points and "how large" it is.

c) Loss Function: The DNN is trained by minimizing the mean squared error (MSE) between the predicted values and their ground truth counterparts. Concretely, the loss function $\mathcal{L}$ combines errors in both the normalized direction vector $\tilde{\mathbf{y}}$ and the scaling factor s:

$$\mathcal{L} = \frac{1}{N}\sum_{i=1}^{N}\left(\left\|\tilde{\mathbf{y}}_i - \hat{\tilde{y}}_i\right\|_2^2 + (s_i - \hat{s}_i)^2\right), \tag{9}$$

where:

- N is the total number of training samples,
- $\tilde{\mathbf{y}}_i$ and s_i are the ground truth direction and scaling factor for the i-th sample,
- $\hat{\tilde{y}}_i$ and $\hat{s}_i$ are the corresponding model predictions.

d) Training Details: The training was conducted in three phases. The initial training was performed for 1700 epochs using the Adam optimizer with a learning rate of 0.001 and a batch size of 32. Earlier runs showed consistent

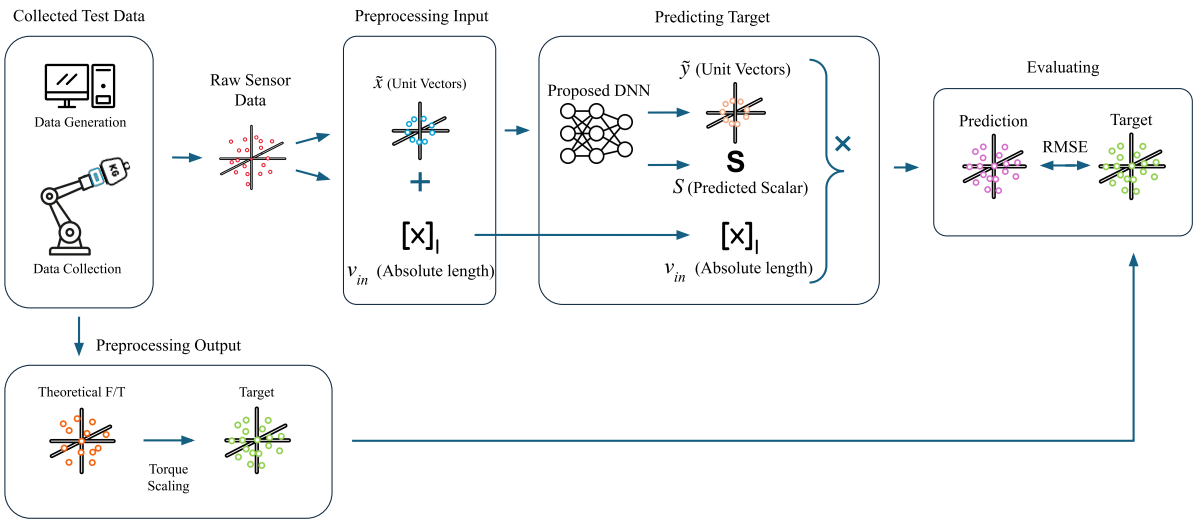


Fig. 2: The prediction process of the proposed DNN.

convergence without over fitting around 1700 epochs. Adam was chosen for its adaptive learning rate capabilities, suitable for optimizing deep networks. This was followed by the first fine-tuning phase, performed for an additional 100 epochs using the Reduce on Plateau method. The learning rate was reduced by a factor of 10 every 30 epochs without improvement, down to a minimum of 0.00001. During the entire learning process, the lowest validation loss was recorded. In the third and last phase, the training continued until the validation loss was within 5% of the lowest recorded loss to achieve a low validation loss. Then the training process stopped. A low validation loss was considered indicative of good performance.

B. Prediction process

After training, the model uses the learned parameters to make predictions on new data. The prediction flow is depicted in Figure 2. The final prediction is constructed by combining the predicted direction vector $\tilde{\mathbf{y}}$ with the scalar s and the raw input magnitude v_{in}. It produces an estimate of the torque-scaled real force-torque in both orientation and size. These prediction can be used to evaluate the performance of the Model. The evaluation metrics used are discussed in Subsection IV-C. The measured force and torque from the torque-scaled values are reconstructed by deviding each torque component by α from Equation 5.

C. Model Architecture

Now that we have discussed the overview of the preprocessing, training, and prediction flows, let's zoom in on the model itself. The architecture of the proposed DNN is designed to capture the nonlinear relationships between the normalized input vectors and the corrected outputs. As illustrated in Figure 3, the model consists of an input layer, multiple hidden layers, and an output layer.

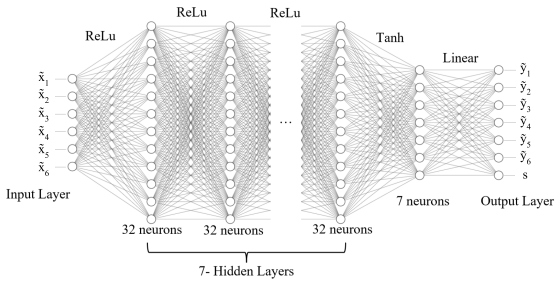


Fig. 3: Architecture of the proposed Deep Neural Network for force-torque sensor calibration. The model inputs the normalized sensor readings and outputs the corrected normalized vector and scaling factor.

The input layer processes the normalized sensor measurements, denoted as $\tilde{\mathbf{x}} \in R^N$, where each component represents a sensor output. Normalizing the inputs ensures consistent scaling, which helps the model learn effectively.

The network includes multiple hidden layers with rectified linear unit (ReLU) activation functions, enabling the model to learn complex nonlinear patterns. The depth and width of the network are chosen based on empirical evaluations to balance model capacity and computational efficiency.

The second-to-last layer uses a hyperbolic tangent (tanh) activation function to normalize the output, ensuring that all values remain within the range $[-1, 1]$, which is suitable for representing directional components.

The output layer has a linear activation function that produces: the corrected normalized vector $\tilde{\mathbf{y}} \in R^6$, representing

the direction of the corrected force-torque measurement, and the scaling factor $s \in R$.

D. Models for Comparison

To demonstrate the effectiveness of our proposed method, we trained two more models for comparison:

a) Comparative DNN Model: As a baseline for comparison, we implemented a traditional Deep Neural Network (DNN) based on the method described in the paper *"Force/torque sensor calibration method by using deep-learning"* [28]. This DNN learns the direct mapping from six-axis capacitance to force/torque output and serves as a benchmark to evaluate the performance of our proposed method. The same baseline model is also utilized in the subsequent study, *"Multi-Axial Force/Torque Sensor Calibration Method Based on Deep-Learning"* [25].

b) Linear Transformation Model (LTM): A linear transformation model using a calibration matrix was also trained. This model represents standard calibration techniques that assume linearity and decoupling of sensor outputs and serves as a baseline for conventional methods.This method has been widely adopted in the literature and is explained in various works, including "Identification of Force-Torque Senso" [29], *"Identification of Force-Torque Sensor Parameters"* [18], and *"Calibration Method for Multi-Axis Force/Torque Sensors"* [19]. These references, among others, show the widespread use and importance of linear transformation models in sensor calibration.

E. Training Procedure of comparative models

Each model had it's own training process.

a) Comparative DNN Model: The model was trained using the Mean Squared Error (MSE) loss function:

$$\mathcal{L} = \frac{1}{N}\sum_{i=1}^{N}\left(\left|F_{x,i}-\hat{F}_{x,i}\right|^2+\left|F_{y,i}-\hat{F}_{y,i}\right|^2+\left|F_{z,i}-\hat{F}_{z,i}\right|^2 + \frac{F_{\max}}{T_{\max}}\left(\left|T_{x,i}-\hat{T}_{x,i}\right|^2+\left|T_{y,i}-\hat{T}_{y,i}\right|^2+\left|T_{z,i}-\hat{T}_{z,i}\right|^2\right)\right) \tag{10}$$

where N is the number of training samples, $F_{x,i}, F_{y,i}, F_{z,i}$ are the true force components, and $\hat{F}_{x,i}, \hat{F}_{y,i}, \hat{F}_{z,i}$ are the predicted force components for the i-th sample. Similarly, $T_{x,i}, T_{y,i}, T_{z,i}$ represent the true torque components, and $\hat{T}_{x,i}, \hat{T}_{y,i}, \hat{T}_{z,i}$ are the predicted torque components. The ratio $\frac{F_{\max}}{T_{\max}}$ ensures that the torque errors, which are smaller in magnitude, are appropriately scaled relative to the force errors.

The training process for the comparative DNN model was largely the same as the proposed model, as detailed in Subsection III-A. The primary difference was that the comparative model converged more slowly, requiring training to continue for 3000 epochs instead of 1700. All other training parameters and fine-tuning phases remained unchanged.

b) Linear Transformation Model (LTM): The LTM was computed using the Least Squares method. This is explained in more detail in Subsection II-B

In the next section, we detail the experimental setup and data used to train and validate our proposed method.

IV. Experimental Setup

In this section, we describe the experimental setup used to create the dataset used to train and evaluate our proposed calibration method. This includes the creation of multiple artificially simulated datasets and the introduction of sensor errors, the creation of a real-world dataset, comparative baseline models and the evaluation metrics used to evaluate the results.

A. Artificial Datasets

We generated 30 artificial datasets to validate the model's robustness and sensitivity. Rather than creating a single dataset with a train/test split, we constructed separate training and testing sets, reflecting how calibration is often done in practice with limited discrete loads. The training set is based on force/torque vectors arranged on four concentric spheres, consistent with common, lower-cost calibration procedures. The test set uses continuously sampled force and torque values across the sensor's operational range, mirroring real-world conditions. Continuing, we generated artificial raw sensor outputs in two steps: first by introducing errors into the force and torque values, then by mapping these values back to sensor readings using the NRS 6200 internal calibration matrix. This approach yields "raw" sensor data that closely simulates realistic outputs.

a) Train set: The training data contained 5000 points, divided evenly across four concentric spheres, with radius steps of 400 N (force) and 8 Nm (torque). These values fall within the Nordbo Robotics sensor's operational ranges of 2000 N and 40 Nm, which are typical for such sensors [30], [31].

b) Test set: The test set consisted of an other 1250 data points. To realize a 20% split relative to the training set. As explained, these points were randomly sampled from the sensor's operational range to capture continuous variations. This approach enables the evaluation of each model's ability to generalize beyond the discrete loads in the training set.

c) Error Simulation: Now that we have the data points errors were introduced: sensor noise, creep (drift), crosstalk (coupling error), and nonlinearity. These errors were applied equally to both the training and test datasets to ensure consistency. The exact error values were sampled randomly with a standard deviation of 10% of the nominal error values, and the axes affected were randomly selected between different datasets. The error values and methods were based on values reported in the literature. An overview of the types of errors can be found in section II-A. Below we explain how the errors were simulated.

Creep (drift): To simulate creep, scaling factors were applied to selected force and torque components. The challenge of modeling creep arises from its time-dependent

or temperature-dependent nature, with limited average or common values reported in the literature. For instance, creep errors in poorly calibrated systems can exceed 0.3% of the measured load [32], whereas well-optimized strain gauge systems have been shown to exhibit creep errors as low as 0.01%–0.02% of the full-scale output [33]. Based on these insights, we adopted a creep error of 0.3%, as a worst case scenario. For a given component (e.g., force F_i), the creep effect was modeled as:

$$F_i' = F_i \times c_{F_i}, \quad c_{F_i} = 1 + \delta_{F_i}, \tag{11}$$

where δ_{F_i} is the creep factor for F_i, sampled from a distribution $\mathcal{N}$ (mean of 0.3%, standard deviation of 0.03%).

Sensor Noise (SN): Gaussian noise was added to each force and torque measurement independently [34]. The standard deviation of the noise was set to 1% of the signal magnitude [35], with the exact value for each sample drawn from a normal distribution with a mean of 1% and a standard deviation of 0.1% (10% of 1%). Mathematically, the noisy wrench vector $\mathbf{w}_{\text{noisy}}$ was computed as:

$$\mathbf{w}_{\text{noisy}} = \mathbf{w} + \mathbf{w} \cdot \boldsymbol{\eta}, \tag{12}$$

where $\mathbf{w}$ is the original wrench vector (forces and torques), and $\boldsymbol{\eta}$ is a noise vector with each component $\eta_i \sim \mathcal{N}(0, \sigma_i^2)$.

Crosstalk (Coupling Error, CE): Studies have demonstrated that crosstalk factors in torque sensors can be reduced to around 5% or lower through optimized sensor designs [36], [37]. To simulate crosstalk in this study, dependencies were introduced between force and torque components, with the affected axes and combinations randomly selected for each dataset. The crosstalk factors were sampled from normal distributions centered around 5% with a standard deviation of 0.5%. The crosstalk from one component to another was modeled as:

$$C_j' = C_j + k_{ij} \times C_i, \tag{13}$$

where C_i and C_j are force or torque components (e.g., F_x, F_y), and k_{ij} is the crosstalk factor sampled as described.

Nonlinearity Error (NE): For six-axis force-torque sensors, reported nonlinearity errors typically range between 0.17% and 1.17% of the full-scale measurement [29], [38], [39], with values depending on design precision and calibration techniques. To simulate nonlinearity, we adopted a worst-case scenario approach, assuming a maximum nonlinearity error of 1.2% of the sensor's full-scale range. This represents a conservative estimate based on the upper bounds of reported values [39]. For this simulation, we used a power-law relationship to model the nonlinearity effect. For any component C_i (force or torque), the nonlinearity was expressed as:

$$C_i' = C_i \left(1 + \gamma \left|C_i\right|^{n-1}\right), \tag{14}$$

where γ is the nonlinearity coefficient and n is the nonlinearity exponent. Based on our assumptions:

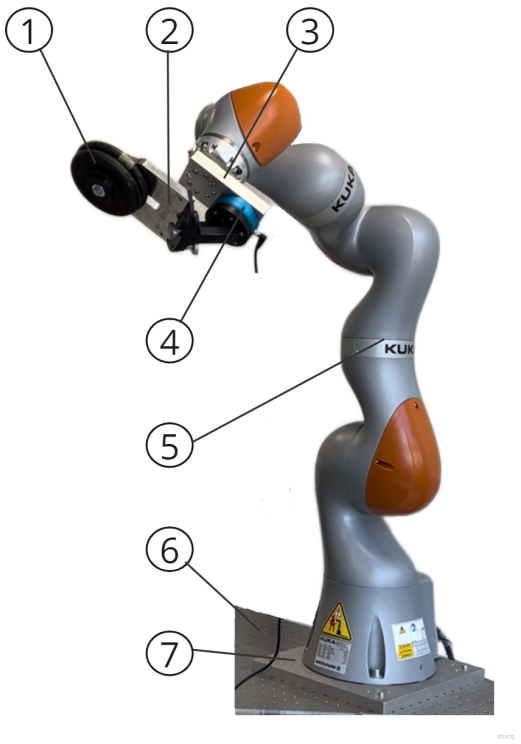


Fig. 4: Overview of the experimental setup with labeled components: (1) external load, (2) mounting device, (3) connection plate, (4) force-torque sensor, (5) KUKA IIWA 14 R820 robotic arm, (6) flat table, and (7) base plate.

- γ is sampled from a normal distribution $\mathcal{N}(\mu, \sigma^2)$, where $\mu = 1.0 \times 10^{-3}$ (corresponding to 1% F.S.) and $\sigma = 1.0 \times 10^{-4}$.
- $n = 2.5$, representing a moderate nonlinearity exponent typically observed in sensor materials [40].

Using these parameters, and within the sensor's defined measurement range, the resulting nonlinearity error aligns with the specified maximum error. This approach ensures that the simulated errors appropriately reflect the worst-case nonlinearity

By incorporating these errors into the datasets, we aimed to achieve a realistic but worst-case scenario error profile for a force-torque sensor based on what the literature suggests.

d) Translating the data into sensor output voltages: In the final step, the internal NRS 6200 calibration matrix, normally used to convert raw sensor readings to force/torque outputs, was inverted and applied in reverse. This transformation mapped the erroneous force/torque vectors into sensor output voltages. The same inverted matrix was used for all datasets. preserving each dataset's unique error profile and generating realistic raw sensor data used for the calibration model.

B. Real-world Dataset

Besides the creation of the artificial data, we also created a "real-world" dataset. This dataset consists of 1,085 data points (7×155), captured by applying seven distinct load configurations from the sensor's reference frame and varying the orientation of the robotic arm in a spherical pattern with the sensor and load attached to its end-effector. This methodology aligns with practical calibration techniques used in industry [31], [30], where discrete known loads and orientations are standard.

a) Experimental Setup: We employed an NRS-6200D80 force-torque sensor (Nordbo Robotics) mounted on a KUKA IIWA 14 R820 robotic arm. As shown in Figure 4, the setup included:

1) External load,
2) Mounting device,
3) Connection plate (30 mm thick aluminum),
4) Force-torque sensor,
5) KUKA IIWA 14 R820 robotic arm,
6) Flat table,
7) Base plate.

b) Load Configurations: To enrich the dataset with a broad range of torque values, seven different load orientations were applied. The center position was set at [186mm, 0mm, 79mm] from the sensor's origin, followed by six additional translations of the external load from this point. Specifically, the load was translated along the three principal axes ($\pm x$, $\pm y$, $\pm z$) relative to that point. The exact translations are given in Table I. Although this particular begin location was initially selected for internal testing, placing the load off-center is essential to diverse torque vectors.

Why Seven Orientations?

- **Single Orientation:** With just the center position load configuration the torque values are confined to a single plane in the 3D torque space.
- **Multiple Orientations** ($\pm x$, $\pm y$, $\pm z$) ensure that all three torque dimensions are excited, producing a richer, more representative calibration dataset.

This variation of the load expands the coverage of torque space and improves the robustness of any calibration or learning-based methods applied thereafter.

c) Robotic Arm Orientations and Spherical Sampling: After fixing a given load configuration, we varied the orientation of the KUKA robotic arm to sample forces and torques from multiple directions. The force vectors correspond to points on a discretized sphere, generated via a *geodesic sphere* approximation [41]. Each point on the unit sphere was mapped to **pitch** and **yaw** angles, which were then converted into joint configurations for the KUKA arm.

Orientations

- **155 Orientations per Sphere:** Each discrete sphere pattern consisted of 155 orientation points, providing a dense yet manageable coverage of force directions.
- **Seven Load Configurations:** Repeating the spherical sampling for each of the seven configurations yielded $7 \times 155 = 1{,}085$ total data points.

d) Reference Force and Torque Calculations: The theoretical force and torque values due to gravity are computed while considering the sensor's orientation in the gravitational frame. Let $\mathbf{R}_{sensor}$ denote the rotation matrix that transforms vectors from the sensor's frame to the gravitational frame. The force vector in the sensor frame is given by:

$$\mathbf{F} = \mathbf{R}_{sensor}^{T}(m\mathbf{g}), \tag{15}$$

where:

- m is the mass of the load,
- $\mathbf{g}$ is the gravitational acceleration vector in the gravitational frame,
- $\mathbf{R}_{sensor}^{T}$ is the transpose (or inverse) of the rotation matrix, which transforms $\mathbf{g}$ into the sensor's frame.

The torque vector in the sensor frame is computed as:

$$\tau = \mathbf{r} \times \mathbf{F}, \tag{16}$$

where $\mathbf{r}$ is the displacement vector from the sensor's origin to the load's center of mass, expressed in the sensor frame.

These theoretical values account for the sensor's orientation and serve as a reference for comparing the sensor's readings.

e) Load vs. No-Load Runs: To account for the mass of the load-holding device (See Figure 4) and isolate the effect of the external load, each spherical run is performed twice. First, in the *no-load* runs, the KUKA's internal gravity compensation is reset, and data are recorded without the external load. Second, for the *load* runs, the external load is attached, gravity compensation is recalculated, and the same spherical pattern of poses is executed again.

Even with careful setup, minor differences (often tenths of a degree) arise between the *no-load* and *load* joint angles. To address this, forward kinematics is used to obtain the end-effector poses $\mathbf{P}_{\text{no-load}}$ and $\mathbf{P}_{\text{load}}$ for pairing joint configurations. A rotation matrix $\mathbf{R}_{\Delta}$ is then derived:

$$\mathbf{R}_{\Delta} = \mathbf{P}_{\text{no-load}}^{\top}\,\mathbf{P}_{\text{load}} \tag{17}$$

which aligns the load configuration's frame to the no-load frame. This transformation is applied to the measured sensor output under load, ensuring that all data share a consistent reference frame.

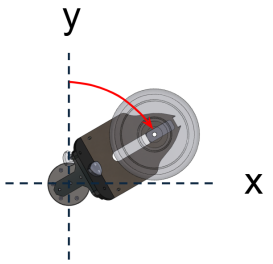
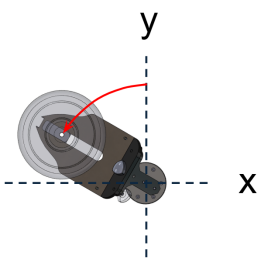
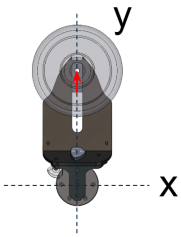
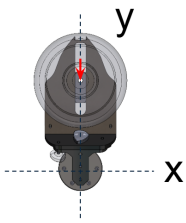
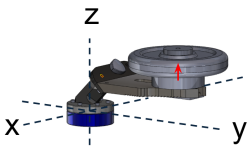
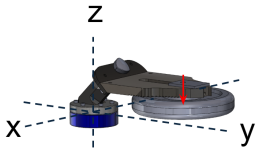
	Positive	Negative	Positive	Negative
Principle Axis: x			x: 161mm y: 93mm z: 0mm	x: -161mm y: 93mm z: 0mm
Principle Axis: y			x: 0mm y: 29mm z: 0mm	x: 0mm y: -42mm z: 0mm
Principle Axis: z			x: 0mm y: 0mm z: 15mm	x: 0mm y: 0mm z: -48mm

TABLE I: The six load configurations with the translations values in mm from the center point at [186mm ,0mm , 79mm]

f) Robot Accuracy.: The documentation of the KUKA robotic arm only specifies the torque accuracy and the positional accuracy of the end effector. The accuracy in terms of joint angles is not provided. Contacting kuka did not achive more clearity, The next part is an assumption made that must not be taken as truth rather as an indication of its rotational accuracy based on the specs of the kuka arm given here [**?**] The KUKA IIWA 14 R820 achieves a positional accuracy of 0.1 mm at the end-effector. In the worst case, if this entire offset were due to the final joint alone (with a link length of about 126 mm), it would correspond to a rotation error of approximately 0.00079 radian. The robotic arm comprises seven individual links, so an accumulated maximum rotational inaccuracy is roughly 0.0055 radian. This could cause an error due to inaccuracy of the measurements of roughly 0.5%. This error is relatively small, but it must be accounted for in the final accuracy assessment. It will only have a minor influence when comparing the performance among different models, as each one is subject to the same error.

C. Evaluation Metrics and Cross-Dataset Evaluation

We use the Root Mean Square Error (RMSE) between the predicted and true values as the primary metric to quantify the performance differences between models. The RMSE is calculated as:

$$\text{RMSE} = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2} \tag{18}$$

where y_i is the true value, $\hat{y}_i$ is the predicted value, and N is the number of samples.

Additionally, we consider the Relative Error of both force and torque separately, to assess the error relative to the magnitude of the true value. This relative error can make the performance more understandable compared to a single RMSE value, which is common in evaluating force torque sensors [11]. The relative errors are calculated as:

$$\text{Relative Error} = \frac{\|y_i - \hat{y}_i\|}{\|y_i\|} \tag{19}$$

Statistical tests, including the Wilcoxon signed-rank test, were used to assess the significance of the differences between models.

For the models specifically, a Cross-Dataset Evaluation was performed to examine the consistency of the models across multiple datasets. This was done by computing the standard deviation of the models' performance metrics and plotting the results of the RMSE in a normal distribution plot.

V. Results

In this section, we first present the performance of the proposed calibration method, the comparative DNN, and

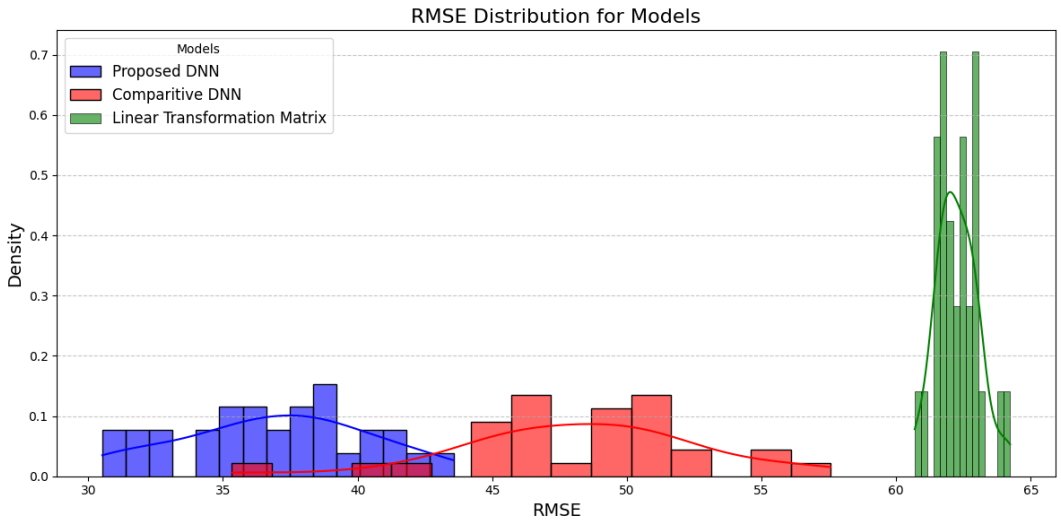


Fig. 5: RMSE Distribution for the Proposed DNN, Comparative DNN, and Linear Transformation Model across Artificial Datasets.

the traditional linear transformation model on artificially generated datasets and then evaluate their performance on real-world sensor data.

A. *Artificial Dataset Results*

The results for the artificial datasets are divided into two parts. First, the RMSE values of the predicted wrench vectors are presented. Each target wrench vector consists of force components and scaled torque components (as shown in Figure 2). These dimensionless values enable a direct comparison of the performance of the three models.

In the second part, the predicted (torque-scaled) wrenches are converted into separate force and torque components. These are then used to calculate the relative errors for force and torque, as discussed in Subsection IV-C.

a) RMSE values of the predicted wrenches: Table II summarizes the average RMSE values across the artificially created datasets for each model.

TABLE II: Average RMSE Results for Artificial Datasets (Mean $\pm$ Std)

Model	RMSE (Mean $\pm$ Std)
Proposed DNN Model	36.9 ± 3.44
Comparative DNN Model	48.3 ± 4.47
Linear Transformation Model	62.3 ± 0.76

The proposed DNN model achieves the lowest average RMSE (approximately 36.9). The comparative DNN model has a higher RMSE of about 48.3, and the linear transformation model trails with an RMSE of approximately 62.3.

Figure 5 illustrates the RMSE distribution for each model across the artificial datasets. The proposed DNN (blue) produces a standard deviation of 3.44, showing slightly better consistency compared to the comparative DNN model (red), which has the highest variability with a standard deviation of 4.47. Both DNN models are remarkably less consistent across datasets compared to the linear transformation model (green), which exhibits the narrowest distribution with a standard deviation of 0.76, reflecting more consistency. This, however, comes at the expense of a higher mean RMSE.

Statistical Significance: A Wilcoxon signed-rank test shows that the differences in RMSE values between the models are statistically significant (p-value < 0.001). This result indicates that the improvements of the proposed DNN model are consistent. However, it is important to note that while this test confirms a statistical difference, it does not provide information on the magnitude of the difference. Traditional parametric effect sizes, like the Cohen's d, could not be calculated, as the differences are not normally distributed.

b) Relative Error of Force and Torque Separately: Below are the results of the relative force and torque errors.

Model	Relative Force Error (Mean $\pm$ Std)	Relative Torque Error (Mean $\pm$ Std)
Proposed DNN	0.040 ± 0.0037	0.038 ± 0.0044
Comparative DNN	0.044 ± 0.0042	0.042 ± 0.0043
LTM	0.070 ± 0.0007	0.062 ± 0.0005

TABLE III: Comparison of Relative Force and Torque Errors (Mean $\pm$ Std) across models.

The proposed DNN has the lowest mean relative errors for both force and torque, with values of 0.040 ± 0.0037 and 0.038 ± 0.0044, respectively. The comparative DNN has slightly higher mean relative errors of 0.044 ± 0.0042 for force and 0.042 ± 0.0043 for torque. The LTM has the highest mean relative errors at 0.070 ± 0.0007 for force and 0.062 ± 0.0005 for torque.

The standard deviations indicate that the LTM exhibits the most consistent performance, with the narrowest distribution of errors for both force and torque. The proposed DNN and comparative DNN models show larger variability across datasets, as indicated by their higher standard deviations.

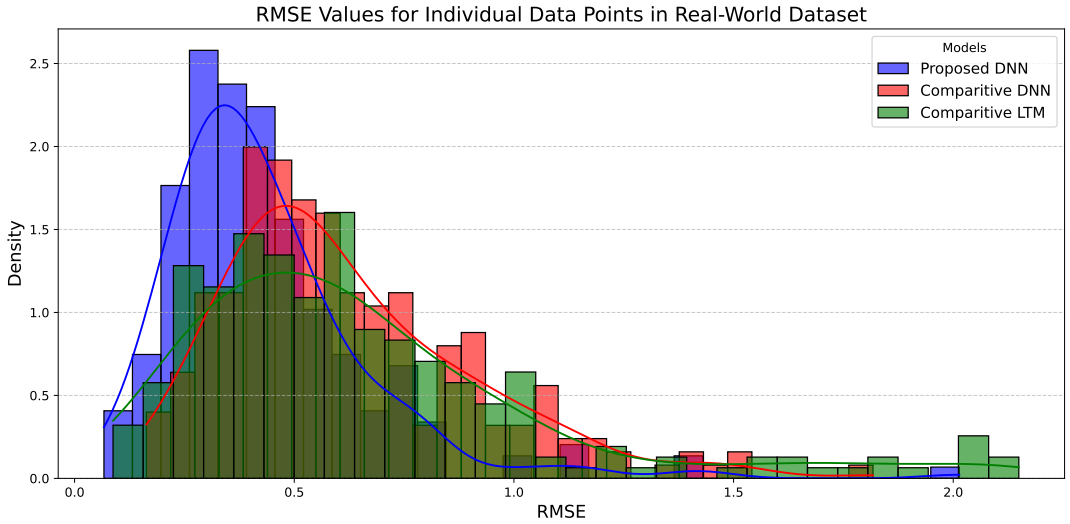


Fig. 6: RMSE Distribution Within the Real-World Dataset for the Proposed DNN, Comparative DNN, and Linear Transformation Model

Among the DNN models, the proposed DNN demonstrates slightly better consistency for both force and torque errors compared to the comparative DNN.

B. Real-World Dataset Evaluation

After establishing the proposed model's performance on artificial datasets, we next evaluated its effectiveness on real-world sensor data.

a) The RMSE of the Predicted Wrenches: Table IV summarizes the RMSE values and standard deviations for each model on the real-world test data.

TABLE IV: RMSE Results for Real-World Data (Mean $\pm$ Std)

Model	**RMSE (Mean $\pm$ Std)**
Proposed DNN Model	0.49 ± 0.24
Comparative DNN Model	0.69 ± 0.29
Linear Transformation Model	0.78 ± 0.42

The proposed DNN model achieves the lowest RMSE on the real-world dataset, with a value of 0.49 ± 0.24. This is lower than the comparative DNN model, which has an RMSE of 0.69 ± 0.29, and significantly better than the linear transformation model, which has the highest RMSE of 0.78 ± 0.42.

The standard deviations reflect the variability of the RMSE values within the single real-world dataset. It is important to note that this variability is distinct from the variability discussed in the artificial dataset section. In the artificial datasets, the variability referred to the consistency of the mean RMSE across multiple datasets (30 datasets). Here, the variability measures how well the models fit the shape of the real-world error within a single dataset. While the two measures are related to performance consistency, they describe different aspects of the models' behavior.

The results indicate that the proposed DNN model not only achieves a lower mean RMSE but also demonstrates more reliable performance within the real-world dataset compared to the other models. The comparative DNN model shows higher mean RMSE and variability, while the LTM exhibits both the highest RMSE and the largest variability within the dataset.

b) Relative Force and Torque Errors: Table V presents the mean relative errors and their standard deviations for each model.

Model	**Relative Force Error (Mean $\pm$ Std)**	**Relative Torque Error (Mean $\pm$ Std)**
Proposed DNN	0.029 ± 0.026	0.022 ± 0.020
Comparative DNN	0.021 ± 0.020	0.045 ± 0.050
LTM	0.041 ± 0.036	0.0090 ± 0.0077

TABLE V: Relative Force and Torque Errors for Real-World Data (Mean $\pm$ Std)

The table shows that for the relative force error, the Comparative DNN model achieves the lowest mean value at 0.021 with a standard deviation of 0.020, compared to (0.029 ± 0.026) for the Proposed DNN model and (0.041 ± 0.036) for the LTM. For the relative torque error, the LTM records the lowest mean at 0.0090 with a standard deviation of 0.0077, while the Proposed DNN model has a mean of (0.0219 ± 0.020) and the Comparative DNN model shows the highest mean at 0.045 with a standard deviation of 0.050.

c) Summary of Findings: Across both artificial and real-world datasets, the proposed DNN calibration method consistently outperforms the comparative DNN model and the traditional Linear Transformation Model. In the artificial datasets, the proposed DNN achieved an average RMSE of 36.9, compared to 48.3 for the comparative DNN model and 62.3 for the Linear Transformation Model. Although the DNN models showed higher variability than the Linear

Transformation Model, the proposed DNN exhibited slightly better consistency with a standard deviation of 3.44 versus 4.47 for the comparative DNN model.

In the real-world sensor data, the proposed DNN again recorded the lowest RMSE, with a mean value of 0.49 and a standard deviation of 0.24, outperforming the comparative DNN and the Linear Transformation Model, which had mean RMSE values of 0.69 and 0.78, respectively. Additionally, the proposed DNN produces the lowest relative force error values, while for torque, the Linear Transformation Model achieves the lowest error, with the proposed DNN remaining competitive.

VI. Discussion

The goal of this work is to develop a calibration model that effectively decouples the coupling effects between axes in force-torque sensors without relying on exhaustive datasets. The model should generalize across the operational range to perform accurately with limited or discrete training data in continuous real-world scenarios. It should also enhance practicality by reducing the need for extensive data collection and simplifying the calibration process.

The experimental results demonstrate the effectiveness of our proposed calibration method for force-torque sensors. By analyzing performance across both real-world and artificial datasets, we observe that the proposed deep neural network (DNN) achieves better measurement accuracy compared to the comparative DNN model and the traditional Linear Transformation Model (LTM). The model consistently keeps the relative force and torque errors under 5%, indicating generally stable accuracy across a range of conditions.

Based on the results of the experiments, the proposed model meets the primary objective of improved accuracy and effective decoupling of inter-axis coupling effects.

These results aligns with the research that is done on DNN in force-torque sensor calibration. The strength of DNNs are shown in several other studies on sensor calibration [26], [12].

a) Critical Points and Limitations: 1. Overall Results of the Calibration The proposed DNN calibration method, despite outperforming the two baseline models, still yields an average error of 3–4%. This error level is high compared to other sensors with accuracy classes of 1% up to 0.1% [42], [43], [44] (i.e., average errors of approximately 1% and 0.1% respectively). Part of this error originates from the experimental setup. For example, the KUKA arm introduces an approximate error of up to 0.5% (see Section IV-B.0.a); note that this value is an approximation based on certain assumptions.

The sensor design may also contribute to the error. Inaccurate strain gauge placement can compromise signal capture, leading to increased crosstalk between axes or reduced sensitivity to applied forces and torques [14]. Additionally, uneven strain distribution across the sensor body further amplifies errors and degrades measurement accuracy [45]. The sensor used in this study is equipped with 12 resistive strain gauges, resulting in a combined raw sensor output from 6 bridges; however, the exact internal design is not publicly accessible.

The calibration method itself also contributes to the overall error. In particular, normalizing input vectors allows the model to generalize from limited data, but it comes at a cost. It also removes magnitude information, preventing the model from capturing intra-axis nonlinearity. These nonlinearities affect the calibration results. According to the sensor's specifications, the maximum intra-axis nonlinear error along the entire operational range is $<4\%$ [46]. These nonlinear behaviors may vary across the sensor's range but are particularly pronounced at its maximum limits, making it challenging to precisely quantify their overall impact on the sensor's total error.

2. Limited Dataset and Sensor Variety: Although these results are promising, they are based primarily on artificially generated data and a single real-world dataset from one specific sensor. This limited dataset may not fully capture the variability present in different sensor designs or operational environments, potentially restricting the generalizability of the findings.

3. Model Variability: The standard deviation values shown in Table II indicate that, although the proposed DNN achieves the best average accuracy, both DNN models show high variability. In contrast, the LTM, though less accurate, is more consistent across datasets. This consistency is likely due to its simpler optimization process, where least squares methods can find a global optimum, whereas DNNs rely on gradient descent, which can become stuck in local minima and is more sensitive to slight variations in training data.

4. Computational Cost: The study did not evaluate the computational cost of predictions made by the DNN or the reconstruction of the final force-torque vector. The LTM, requiring significantly fewer calculations per measurement, can operate effectively on much smaller processors compared to the deep neural network. In some cases, the computational demands of the DNN may be too high to achieve the necessary measurement frequency. However, in certain applications the improved accuracy can be more beneficial, for example in post-procedure assessment tasks.

5. Model Development Process: While the proposed DNN demonstrates strong generalization capabilities and improved accuracy compared to baseline models, the development process relied heavily on empirical methods and trial-and-error approaches for hyperparameter selection, architecture design, and training strategies. No systematic optimization was performed to ensure that the chosen architecture or hyperparameters are globally optimal.

b) Future Perspectives: Future work should include a more structured approach to model optimization, such as hyperparameter tuning with grid search or Bayesian optimization, as well as experiments with alternative architectures or regularization techniques. These efforts could potentially reduce the variability observed in DNN performance and ensure that the model achieves its full potential. Evaluating the computational cost of DNN predictions and the reconstruction of the final force-torque vector is also necessary to

ensure suitability for real-time applications.

Other future work could focus on refining the experimental setup, particularly by enhancing the calibration of the KUKA robotic arm to further reduce measurement error. Several studies have proposed techniques to improve the accuracy of the KUKA robotic arm by as much as fourfold [47], [48]. Incorporating these advanced calibration techniques could mitigate error contributions from the experimental setup, thereby enhancing overall sensor performance.

Finally, applying the model to a broader range of sensor designs and multiple sensors will help validate its performance more robustly, as each design and individual sensor exhibits unique characteristics. Expanding the validation to include diverse sensor types and operational conditions will provide a more comprehensive assessment of the model's generalizability and help identify any sensor-specific challenges.

VII. Conclusion

This work presented a novel calibration model based on a deep neural network (DNN) that effectively decouples inter-axis effects in force-torque sensors using limited, discrete training data. Our experiments, conducted on both real-world and artificial datasets, demonstrate that the proposed DNN achieves superior measurement accuracy compared to both a comparative DNN model and the traditional Linear Transformation Model (LTM), consistently keeping relative errors under 5%. This marks a significant improvement in sensor calibration, especially in terms of decoupling inter-axis effects without the need for exhaustive datasets.

However, our results also shows several limitations. The average error of 3–4% remains high when compared to sensors with accuracy classes between 1% and 0.1%, partly due to factors such as experimental setup imperfections, inherent sensor design issues, and calibration method constraints. Additionally, the higher computational cost of the DNN, compared to simpler linear models, suggests that its deployment may be more suited to applications where post-procedure assessment is acceptable rather than real-time measurement.

Future work will focus on systematic model optimization, refining the calibration process, and exploring advanced techniques to mitigate experimental errors, especially those stemming from the KUKA robotic arm. Broadening the evaluation to include a wider range of sensor designs will also help in validating and generalizing the effectiveness of our approach.

References

[1] A. Shademan, R. S. Decker, J. Opfermann, S. Léonard, A. Krieger, and P. C. W. Kim, "Supervised autonomous robotic soft tissue surgery," *Science Translational Medicine*, vol. 8, pp. 337ra64 – 337ra64, 2016.

[2] Y. Kim, U. Kim, D. Seok, J. So, Y. Lee, and H. Choi, "Torque sensor embedded actuator module for robotic applications," *IEEE/ASME Transactions on Mechatronics*, vol. 23, pp. 1662–1672, 2018.

[3] A. Wahrburg, J. Bös, K. D. Listmann, F. Dai, B. Matthias, and H. Ding, "Motor-current-based estimation of cartesian contact forces and torques for robotic manipulators and its application to force control," *IEEE Transactions on Automation Science and Engineering*, vol. 15, pp. 879–886, 2018.

[4] U. Kim, D.-H. Lee, Y.-B. Kim, D.-Y. Seok, and H. R. Choi, "A novel six-axis force/torque sensor for robotic applications," *IEEE/ASME Transactions on Mechatronics*, vol. 22, no. 3, pp. 1381–1391, 2017.

[5] Y. Noh *et al.*, "Multi-axis force/torque sensor based on simply-supported beam and optoelectronics," *Sensors*, vol. 16, no. 11, 2016.

[6] D.-H. Lee, U. Kim, H. Jung, and H. R. Choi, "A capacitive-type novel six-axis force/torque sensor for robotic applications," *IEEE Sensors Journal*, vol. 16, no. 8, pp. 2290–2299, 2016.

[7] A. Song, J. Wu, G. Qin, and W. Huang, "A novel self-decoupled four degree-of-freedom wrist force/torque sensor," *Measurement*, vol. 40, no. 9, pp. 883–891, 2007.

[8] M.-K. Kang, S. Lee, and J.-H. Kim, "Shape optimization of a mechanically decoupled six-axis force/torque sensor," *Sensors and Actuators A: Physical*, vol. 209, pp. 41–51, Mar. 2014.

[9] C. Ding, Y. Han, W. Du, J. Wu, and Z. Xiong, "In situ calibration of six-axis force–torque sensors for industrial robots with tilting base," *IEEE Transactions on Robotics*, vol. 38, no. 4, pp. 2308–2321, 2021.

[10] Y. Hou, J. Yao, D. Zeng, J. Chen, and Y. Zhao, "Development and calibration of a hyperstatic six-component force/torque sensor," *Chinese Journal of Mechanical Engineering*, vol. 22, pp. 505–513, 2009.

[11] M. Y. Cao, S. Laws, and F. R. y. Baena, "Six-axis force/torque sensors for robotics applications: A review," *IEEE Sensors Journal*, vol. 21, no. 24, pp. 27 238–27 251, 2021.

[12] D. J. Rodriguez, J. J. Castañeda, S. P. Mejía, A. P. Torres, R. Carelli, and P. Fiorini, "Using deep neural networks to improve contact wrench estimation of serial robotic manipulators in static tasks," *Frontiers in Robotics and AI*, vol. 9, p. 892916, 2022.

[13] Y.-J. Wang and C.-W. Tung, "A machine for calibrating six-axis force/torque sensors using a torque sensing structure and a stiffness compensation model," *Sensors and Actuators A: Physical*, vol. 334, p. 113323, 2022.

[14] Y.-J. Wang, C.-W. Hsu, and C.-Y. Sue, "Design and calibration of a dual-frame force and torque sensor," *IEEE Sensors Journal*, vol. 20, pp. 12 134–12 145, 2020.

[15] Y. Ma, S. Xie, X. Zhang, and Y. Luo, "Hybrid calibration method for six-component force/torque transducers of wind tunnel balance based on support vector machines," *Chinese Journal of Aeronautics*, vol. 26, pp. 554–562, 2013.

[16] O. Al-Mai and M. Ahmadi, "Novel calibration methodologies for compliant, multiaxis, fiber-optic-based force/torque sensors," *IEEE Sensors Journal*, vol. 22, pp. 21 727–21 734, 2022.

[17] C.-Y. Lin, A. R. Ahmad, and G. A. Kebede, "Novel mechanically fully decoupled six-axis force-moment sensor," *Sensors (Basel, Switzerland)*, vol. 20, 2020.

[18] S. Traversaro, D. Pucci, and F. Nori, "In situ calibration of six-axis force-torque sensors using accelerometer measurements," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2015, pp. 2111–2116.

[19] F. J. A. Chavez, S. Traversaro, D. Pucci, and F. Nori, "Model based in situ calibration of six axis force torque sensors," in *Proc. IEEE-RAS 16th Int. Conf. Humanoid Robots*, 2016, pp. 422–427.

[20] D. Ming, X. Zhang, X. Liu, B. Wan, Y. Hu, and K. D. K. Luk, "Nonlinear static decoupling of six-dimension force sensor for walker dynamometer system based on artificial neural network," in *Proceedings of the International Conference on Computational Intelligence and Measurement Systems Applications (CIMSA)*, 2009, pp. 14–17.

[21] H. Cao, Y. Yu, and Y. Ge, "A research of multi-axis force sensor static decoupling method based on neural network," in *Proceedings of the International Conference on Automation and Logistics (ICAL)*, 2009, pp. 875–879.

[22] J. Ma, A. Song, and J. Xiao, "A robust static decoupling algorithm for 3-axis force sensors based on coupling error model and -svr," *Sensors*, vol. 12, no. 11, pp. 14 537–14 555, 2012.

[23] J. Ma and A. Song, "Fast estimation of strains for cross-beams six-axis force/torque sensors by mechanical modeling," *Sensors*, vol. 13, no. 5, pp. 6669–6686, 2013.

[24] T.-F. Lu, G. C. I. Lin, and J. R. He, "Neural-network-based 3d force/torque sensor calibration for robot applications," *Engineering Applications of Artificial Intelligence*, vol. 10, no. 1, pp. 87–97, 1997.

[25] H. S. Oh, U. Kim, G. Kang, J. K. Seo, and H. Choi, "Multi-axial force/torque sensor calibration method based on deep-learning," *IEEE Sensors Journal*, vol. 18, pp. 5485–5496, 2018.

[26] J. Xia, H. Zhang, Y. Wu, C. Liu, and S. Li, "Deep learning-based sensor calibration: A review of methods, applications, and future perspectives," *Frontiers in Robotics and AI*, vol. 9, 2022.

[27] D. Chen, A. Song, and A. Li, "Design and calibration of a six-axis force/torque sensor with large measurement range used for the space manipulator," *Procedia Engineering*, vol. 99, pp. 1164–1170, 2015, 2014 Asia-Pacific International Symposium on Aerospace Technology, APISAT2014 September 24-26, 2014 Shanghai, China. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1877705814038168

[28] H. S. Oh, G. Kang, U. Kim, J. K. Seo, W. You, and H. Choi, "Force/torque sensor calibration method by using deep-learning," *IEEE Sensors Journal*, pp. 777–782, 2017.

[29] Q. Liang, D. Zhang, Q. Song, Y. Ge, H. Cao, and Y. Ge, "Design and fabrication of a six-dimensional wrist force/torque sensor based on e-type membranes compared to cross beams," *Measurement*, vol. 43, no. 10, pp. 1702–1719, 2010.

[30] Robotiq, *FT Sensor Instruction Manual*, 2018, accessed: 2024-07-12.

[31] A. I. Automation, *Axia130 RS422 Force/Torque Sensor Manual*, 2023, accessed: 2024-07-13.

[32] B. Slutsky, "Compensation of anelastic error in force measurement," *AIAA Journal*, vol. 43, no. 6, pp. 1326–1335, 2005.

[33] M. Mathis, D. Vollberg, M. Langosch, D. Göttel, A. Lellig, and G. Schultes, "Creep adjustment of strain gauges based on granular nicr-carbon thin films," *Journal of Sensors and Sensor Systems*, vol. 10, no. 1, pp. 53–61, 2021.

[34] S. G. Kwak and J. H. Kim, "Central limit theorem: the cornerstone of modern statistics," *Korean journal of anesthesiology*, vol. 70, no. 2, pp. 144–156, 2017.

[35] A. H. H. Hosseinabadi, D. G. Black, and S. E. Salcudean, "Ultra low-noise fpga-based six-axis optical force–torque sensor: Hardware and software," *IEEE Transactions on Industrial Electronics*, vol. 68, no. 10, pp. 10 207–10 217, 2020.

[36] J.-K. Min and J.-B. Song, "Sensor block type joint torque sensor insensitive to crosstalk error," *IEEE Sensors Journal*, vol. 20, pp. 3469–3475, 2020.

[37] S. yun Choi, T.-K. Kim, D. Y. Kim, B.-S. Kim, J.-H. Hwang, and C.-W. Park, "Development of joint torque sensor applied to compensate crosstalk error," in *2012 IEEE International Conference on Automation Science and Engineering (CASE)*, 2012, pp. 1086–1088.

[38] C.-C. Lin, C.-Y. Su, S.-T. Lin, C.-Y. Chen, C.-N. Yeh, C.-H. Lin, L.-W. Wang, S.-Y. Kuo, and L.-J. Chien, "6-dof force/torque sensor," *2019 14th International Microsystems, Packaging, Assembly and Circuits Technology Conference (IMPACT)*, pp. 191–194, 2019.

[39] Z. Yongsheng, "Dual layers pre-stressed six-axis force sensor and its static calibration," *Journal of Mechanical Engineering*, 2013.

[40] L. Fu, A. Song, and D. Chen, "A polyetheretherketone six-axis force/torque sensor," *IEEE Access*, vol. 7, pp. 105 391–105 401, 2019.

[41] B.-Y. Chen and L. Vanhecke, "Differential geometry of geodesic spheres." *Surgical endoscopy*, 1981.

[42] H. H. B. Kjær, *MCS10 Multi-Component Sensor for Complex Load Analysis*, n.d. [Online]. Available: https://www.hbkworld.com/en/products/transducers/multi-component/mcs10

[43] P. Instrumentation, *K6D130 6-Axis Force/Torque Transducer - 1kN200Nm to 15kN12kNm*, n.d. [Online]. Available: https://en.pm-instrumentation.com/k6d130-6-axis-forcetorque-transducer-1kn200nm-to-15kn12knm

[44] ——, *Datasheet F6D80e - 6-Axis Force/Torque Sensor*, n.d. [Online]. Available: https://www.me-systeme.de/en/f6d80-40e-300n

[45] Y. Sun, "Design, manufacture, test and experiment of six-axis force torque sensor for chinese experimental module manipulator," *Sensors (Basel, Switzerland)*, vol. 22, no. 9, p. 3603, 2022. [Online]. Available: https://doi.org/10.3390/s22093603

[46] "Nordbo Robotics Workshop." [Online]. Available: https://workshop.nordbo-robotics.com/documentation/force-torque-sensors/nrs-6-xxx-dxx-documentation

[47] K. V. Wyk, J. Falco, and G. Cheok, "Efficiently improving and quantifying robot accuracy in situ," *IEEE Transactions on Automation Science and Engineering*, vol. 1, 2019.

[48] M. Sumanas, A. Petronis, V. Bucinskas, A. Dzedzickis, D. Virzonis, and I. Morkvenaite-Vilkonciene, "Deep q-learning in robotics: Improvement of accuracy and repeatability," *Sensors*, vol. 22, no. 10, 2022. [Online]. Available: https://www.mdpi.com/1424-8220/22/10/3911