Physics–Guided Neural Networks for Feedforward Control: with application to an industrial linear motor

Nard Strijbosch[1,2], Kaan Çalık1[1,2], Max Bolderman[2], Mircea Lazar[2] and Theresa Spaan-Burke[1]
[1] IBS Precision Engineering, Eindhoven, The Netherlands
[2] Eindhoven University of Technology, Eindhoven, The Netherlands strijbosch@ibspe.com

Keywords: Artificial intelligence (AI), Machine learning (ML), Feedforward control, Physics-based neural networks (PGNNs), High precision measurement machines

Original publication: DSPE Conference on Precision Mechatronics - 26 September 2023

1 Artificial intelligence and high precision measurement machines

Artificial intelligence (AI) has emerged as a transformative technology, revolutionizing various industries and domains. One area which could benefit significantly from AI is the field of high precision measurement machines. These machines play a crucial role in industries such as aerospace, semiconductors, and automotive, where accurate and reliable measurements are essential for quality assurance, process optimization, and innovation. A few examples of measurement machines developed by IBS Precision Engineering are depicted in Figure 1

One specific area in measurement machines where AI can make significant contributions is motion control. AI techniques, such as artificial neural networks, can be leveraged to build models that capture the complex relationships between disturbances and control actions, enabling more accurate and predictive feedforward control inputs [1]. These models can learn from large datasets and adapt over time, allowing the system to continuously improve its ability to anticipate and compensate for disturbances.

By combining AI with feedforward control, systems can achieve .faster settling while reducing tracking errors, ultimately leading to increased throughput and accuracy. Although AI models excel at pattern recognition, their modelling process is often considered a "black box," making it difficult to fully understand the underlying physics an artificial neural network is modelling. This is a major drawback of these models when implementing them in industrial applications. This paper presents a novel AI approach for feedforward control that is guided by a physics-based model. This includes the following contributions:

- C1. A high level of interpretability utilizing a model based on first principal physics.
- C2. An artificial neural network that only captures the hard-to-model dynamics.
- C3. Continuous updates of the AI model to track changes in the dynamics due to, e.g. varying measurement objects or wear and tear.
- C4. An experimental validation on an industrial linear motor that illustrates the potential of this feedforward approach.

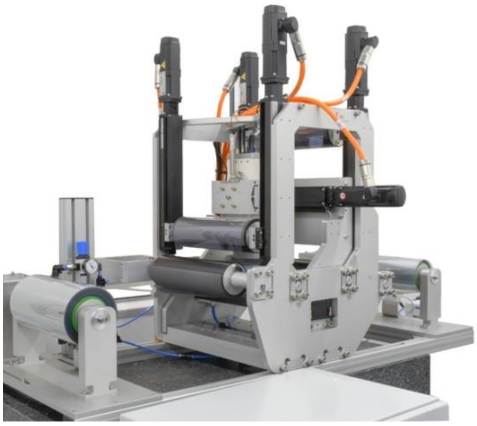

Figure 1 High precision machines developed at IBS precision engineering. Left: the ISARA 400 a 3D coordinate measurement machine with 11nm uncertainty for freeform optics. Right: In-line interferometry system with motion stages that ensure an area of interest on continuously moving foil is stationary with respect to an interferometer to measure nanometer features on photovoltaic foils, more info see [9].

2 The potential of a perfect feedforward controller

A typical motion control architecture is shown in Figure 2. Here, *P*, is a positioning system, in this case a linear motor, with a force, *u*, as input signal and a position measurement, *y*, as output. The goal of the control system is to minimize the error, *e*, between a desired trajectory y_d and the measured output of the system *y*. The system should be capable to track various trajectories y_d in the presence of various unreproducible disturbances, e.g., floor vibrations, as illustrated by *d*. To achieve this a feedback controller and a feedforward controller are used.

The goal of the feedforward controller *F* is to compensate for all predictable disturbances based on the knowledge of the desired trajectory, i.e., the perfect feedforward controller *F* would lead to zero error in the absence of disturbances, *d*. The feedback controller is used to suppress the impact of any disturbances *d* and the remaining error due to imperfections in the design of *F*.

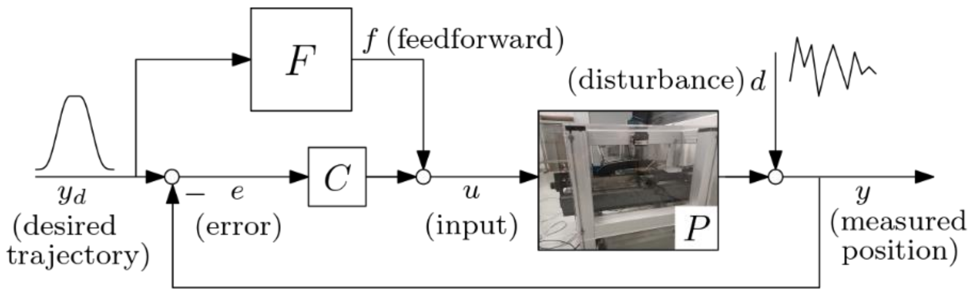

Figure 2 Motion control architecture.

The goal of the feedforward controller *F* is to compensate for all predictable disturbances based on the knowledge of the desired trajectory, i.e., the perfect feedforward controller *F* would lead to zero error in the absence of disturbances, *d*. The feedback controller is used to suppress the impact of any disturbances *d* and the remaining error due to imperfections in the design of *F*.

Figure 3 illustrates the highly reproducible nature of the positioning system depicted in Figure 2. Here 6 identical experiments are performed without any feedforward and the y_d as depicted in Figure 2. The tracking error for each experiment shows a high reproducibility. When removing the average the non-repeating part of the tracking error remains, as depicted in Figure 4. This indicates that when applying a feedforward controller that is capable of predicting the required forces to compensate for the reproducible part, the tracking error can be reduced from 1000 μm to a maximum error of only 7.5 μm.

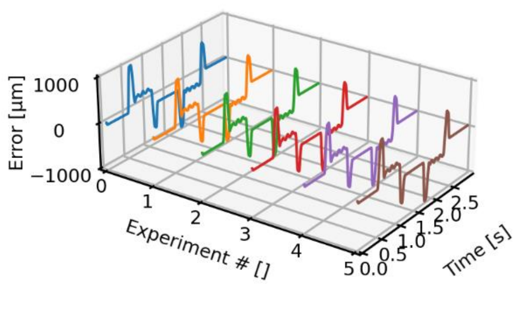

Figure 3 Highly reproducible errors for identical motion tasks.

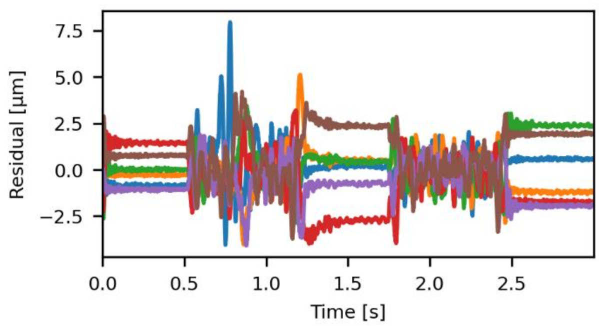

Figure 4 Non-repeating part of the tracking error for identical motion tasks.

3 A model based feedforward controller

A typical feedforward controller is designed based on physical knowledge of the system. For example, the linear motor as depicted in Figure 2 consists of a mass, hence, a force proportional to the desired acceleration is required. Moreover, friction is present, i.e. forces proportional to the velocity and direction are used to model viscous friction and Coulomb friction. This physics-based feedforward control design is depicted in Figure 5 and is mathematically given by

$$f = \theta^m \cdot \ddot{y}_d + \theta^v \cdot \dot{y}_d + \theta^c \cdot \mathrm{sign}(\dot{y}_d)$$

with θ^m [kg] the mass, θ^v[Ns/m] the viscous friction coefficient, and, θ^c [N] the coulomb friction coefficient. These parameters can be tuned manually or automatically [2].

This physics based model leads to high confidence in the performance of the feedforward controller on any trajectory, thereby enabling implementation in industry. The main drawback of employing a physics based feedforward controller is that the performance it can achieve is limited by the accuracy of the model, in other words when the performance does not reach the level of the reproducibility of the system a more extensive model of the system is required. Feedforward controllers exist to compensate for more complex dynamic effects like hysteresis, see, e.g., [3]. However, an extensive time investment is required to model these complex dynamics which yields only a limited performance increase.

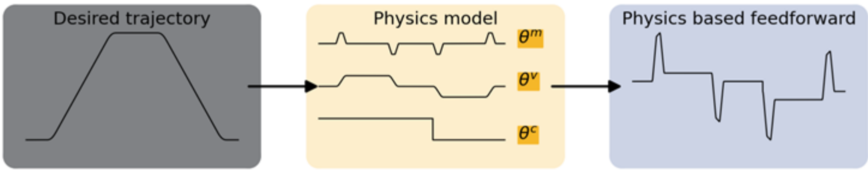

Figure 5 Model based feedforward control strategy.

4 Artificial Neural Networks: universal function approximators

Artificial Neural Networks (ANNs) are powerful mathematical models that can approximate complex functions. ANNs can be thought of as networks of interconnected "neurons" that work together to solve problems. These neurons are inspired by the ones in human brains, and they help ANNs understand and learn patterns in data.

One fascinating feature of ANNs is their ability to approximate any kind of function with enough training. This means that, given the right setup and enough computational power, ANNs can perfectly model the system that generated the data. Exploiting an ANN that models the dynamics of a system beyond first principals of physics as a feedforward controller could enable a breakthrough in performance without the need to model complex dynamics.

In Figure 6 a schematic representation of an ANN is presented, each dot is referred to as a neuron. An individual neuron in an artificial neural network serves as the fundamental building block for information processing. It mimics the behaviour of a biological neuron, albeit in a simplified manner. The neuron receives inputs from other neurons or external sources, each of which is multiplied by a corresponding weight. These weighted inputs are then summed together, and an activation function is applied to the sum. The activation function introduces non-linearity, allowing the neuron to model complex relationships in the data. The resulting output is passed on to other neurons as inputs. By adjusting the weights and biases during the learning process, the neuron can adapt and refine its responses, enabling the neural network to learn and make predictions based on the given inputs.

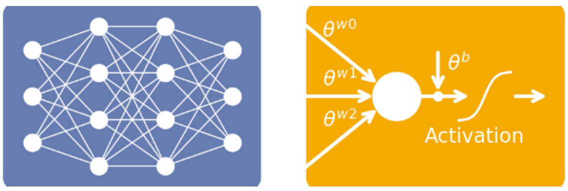

Figure 6 Schematic representation of an artificial neural network (left), and a single neuron (right).

Since obtaining guarantees on the output of an ANN is hard, replacing the complete feedforward controller with an ANN is undesired. However to take advantage of ANNs in feedforward control in this paper the physics guided neural network as depicted in Figure 7 is considered, [4] [5] [6]. This feedforward controller exploits a physics based model like a conventional feedforward controller, in addition a neural network is used to model all the remaining dynamics.

The goal of this feedforward controller structure is to rely on the physics based part to obtain a high level of interpretability and a high confidence of applying the feedforward controller on any desired trajectory, leading to Contribution C1. The ANN is only used to compensate for the remaining part of the dynamics, thereby enabling to compensate for the complete reproducible part of the error while having only a limited force predicted by the "black box" ANN compared to the physics based feedforward, leading to Contribution C2.

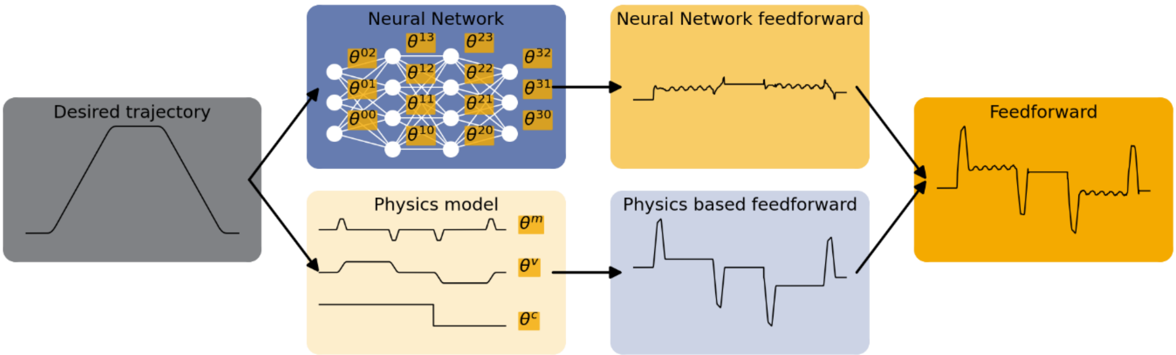

Figure 7 Physics guided neural network based feedforward control strategy.

5 Training an artificial neural network

To fully exploit the potential of a PGNN-based feedforward controller all the parameters need to be optimally tuned at any time. Finding these optimal parameters is often referred to as training the neural network. The traditional neural network training process consists of the following steps:

1. Gathering a large dataset with desired trajectories representative for normal system operation.
2. Solve a large non convex optimization problem to find the parameters that fit the data best.

Step 1 is trivial, since most industrial positioning systems are equipped with sufficient data acquisition tools. Availability of sufficient measurement time is the only challenge.

Recent developments in data science enable a straightforward implementation of Step 2. Open-source software like Tensorflow [7] provides various optimization algorithms that aim to find the optimal values for the parameters of a neural network given a performance criterion.

6 Continuous Learning

To enable a continuously improving feedforward controller this paper proposes a training procedure which iteratively updates the parameters of a PGNN with small updates based on measured data during normal operation. These continuous updates enable the PGNN to track slow changing dynamics, e.g., due to wear of the system, without the need to fully retrain the PGNN, this constitutes Contribution C3.

The proposed procedure is given below and is inspired by common automated feedforward tuning procedures for feedforward controllers [2], [8].

For each iteration j:

1. Perform a set of representative experiments and gather the tracking error e_j with the feedforward controller parameterized by the set of parameters θ_j. The desired trajectories can vary for each iteration, i.e., data gathering during normal operation is possible.
2. Based on the tracking error e_j and an approximate model of the system, predict the tracking error e_{j+1} as function of the to be determined parameters θ_{j+1}. An example of such an approximation is
$$\hat{e}_{j+1}(\theta_{j+1}) = e_j - \widehat{SP} f_j(\theta_j) + \widehat{SP} f_{j+1}(\theta_{j+1})$$
here $\widehat{SP} = (1 + \hat{P}C)^{-1}\hat{P}$ with $\hat{P}$ an approximate model of the system and C the feedback controller.
3. Determine the values for θ_{j+1} that optimize the predicted tracking performance, i.e.,

$$\theta_{j+1} = \text{argmin}_{\theta_{j+1}} \|\hat{e}_{j+1}(\theta_{j+1})\|$$

To enforce a desired behaviour during the training procedure extra penalty's could be added to the optimization problem of step 3. Examples of such penalties are:

P1. $\|\theta_j^m - \theta_0^m\|$ with θ_0^m the initial estimate of the mass of the system. This penalty ensures that the physics guided part will stay close to the original estimate. This prevents the neural network to also model the mass leading to an incorrect interpretability of the physics guided mass parameter. More information about penalties that ensure interpretability of the physics guided part, see [5], [6].

P2. $\|f_j - f_{j+1}\|$ to penalize large changes of the feedforward controller, e.g., when a large non reproducible disturbance is present during an iteration this should not be compensated for by the feedforward controller in the next iteration.

7 Experimental validation on an industrial linear motor

To validate the performance improvement that can be obtained by applying a PGNN the linear industrial motor as given in Figure 2 is used to compare various feedforward control strategies. The following feedforward control strategies are compared:

A. A physics based feedforward controller that compensates for: mass, viscous friction, Coulomb friction, a constant force from the cable slab and a sinusoidal force ripple periodic with the magnet pitch. The parameters are automatically tuned following [2].

B. A PGNN that is trained following the procedure described in the previous section with only a single iteration including only a Penalty P1 for each physical parameter, this is similar to the approach in [5].

C. An iteratively trained PGNN with a Penalty P1 for each physical parameter and a Penalty P2.

The PGNN structures used by approach B and C are identical. The PGNN contains a physics guided part identical to the feedforward controller of approach A and a neural network with 2 hidden layers which each contain 16 neurons with hyperbolic tangent activation functions. Each of the feedforward controllers is trained on a data set consisting of 20 different representative desired trajectories of 3 seconds sampled with 1 kHz. In each of the training iterations of the feedforward controller 8 random trajectories from this set are used instead of all 20. The performance of each strategy is validated based on an additional desired trajectory which is not part of the training trajectories. This results in the performance comparison presented in Figure 9, which indicates a superior performance of the iteratively updated PGNN.

The tracking error for the validation trajectory for each iteration of approach C is given in Figure 8. Moreover for each approach the final tracking error and feedforward signal for the validation trajectory is given in Figure 10 and Figure 11, respectively. Figure 10 highlights the obtained performance increase of exploiting a PGNN compared to only a physics based feedforward controller, while Figure 11 shows that the contribution of the ANN to the total feedforward is only minimal.

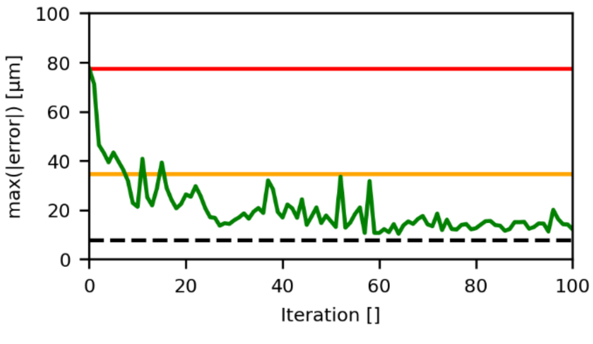

Figure 9 Performance comparison for various feedforward control strategies: physics based (−), single training PGNN (−), iteratively trained PGNN (−). Dashed: the nonrepeating part of the error

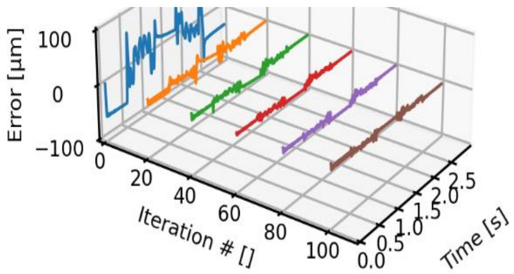

Figure 8 Evolution of the tracking error over the iterations for the validation trajectory with the iteratively updated PGNN based feedforward controller.

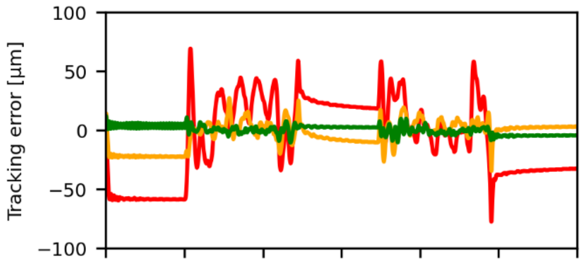

Figure 10 The remaining tracking error after training for each of the feedforward control strategies: physics based (−), single training PGNN (−), iteratively trained PGNN (−).

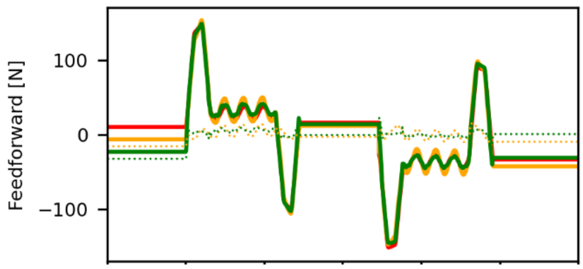

Figure 11 The obtained feedforward signal after training for each of the feedforward control strategies: physics based (−), single training PGNN (−), iteratively trained PGNN (−). The predicted force of each ANN is given by dotted lines.

References

[1] L. Aarnoudse, W. Ohnishi, M. Poot, P. Tacx, N. Strijbosch and T. Oomen, “Control-relevant neural networks for intelligent motion feedforward,” pp. 1-6, March 2021.

[2] S. Van der Meulen, R. Tousain and O. Bosgra, “Fixed structure feedforward controller design exploiting iterative trials: Application to a wafer stage and a desktop printer,” *Journal of Dynamic Systems, Measurement and Control : Transactions of the ASME*, vol. 130, pp. 051006--1/16, 2008.

[3] N. Strijbosch, E. Verschueren, K. Tiels and T. Oomen, “High Precision Sample Positioning in Electron Microscopes - Motion Feedforward Tuning for Hysteretic Piezo Actuators,” *Mikroniek*, vol. 4, pp. 26-31, 2021.

[4] M. Bolderman, M. Lazar and H. Butler, “Physics–Guided Neural Networks for Inversion–based Feedforward Control applied to Linear Motors,” *CCTA 2021 - 5th IEEE Conference on Control Technology and Applications*, pp. 1115-1120, 2021.

[5] M. Bolderman, M. Lazar, H. Butler, S. Koekebakker, E. van Horssen, R. Kamidi, T. Spaan-Burke and N. Strijbosch, “Physics-guided neural networks for feedforward control with input-to-state stability guarantees,” *arXiv preprint arXiv:2301.08568*, 2023.

[6] J. Kon, D. Bruijnen, v. d. J. Wijdeven, Heertjes, M. and T. Oomen, “Physics-guided neural networks for feedforward control: An orthogonal projection-based approach,” *2022 American Control Conference (ACC)*, pp. 4377-4382, June 2022.

[7] M. Abadi and et al., “TensorFlow: Large-scale machine learning on heterogeneous systems,” *Software available from tensorflow.org*, 2016.

[8] N. Strijbosch, L. Blanken and T. Oomen, “Frequency domain design of iterative learning control and repetitive control for complex motion systems,” 2018.

[9] T. Spaan-Burke, “Dances with Foils,” *Mikroniek*, no. 3, pp. 16-20, 2022.