(Machine) learning for feedforward in precision mechatronics

DSPE YOUR PRECISION PORTAL
2023 CONFERENCE ON PRECISION MECHATRONICS
26 & 27 September
Sint Michielsgestel
www.dspe.nl/conference

(Machine) learning for feedforward in precision mechatronics

Leontine Aarnoudse[1], Tom Oomen[1,2]
[1] Eindhoven University of Technology, Eindhoven, The Netherlands
[2] Delft University of Technology, Delft, The Netherlands
l.i.m.aarnoudse@tue.nl

Extended abstract for poster
High tracking performance for mechatronic systems requires accurate feedforward control, which can be learned from data through dedicated efficient algorithms. This research aims to improve performance through new approaches at the intersection of machine learning (neural networks, random learning), controls (feedforward), and precision mechatronics. Four examples are: 1) automated model-free feedforward tuning, 2) fast and accurate iterative learning control (ILC) without noise amplification through nonlinear filters, 3) the use of control-relevant neural networks for feedforward, and 4) ILC method for accurate contour tracking.

1) Randomized experiments lead to efficient learning of MIMO feedforward parameters

Accurate parameterized feedforward control is at the basis of many successful control applications with varying references. However, parameterized feedforward control for MIMO systems with interaction involves many interdependent parameters, such that manual tuning approaches are infeasible in practice. In addition, due to modelling and design requirements the application of automated feedforward tuning approaches such as ILC are not trivial. Therefore, this research aims to develop an efficient data-driven approach to learn the feedforward parameters for MIMO systems.

Approach: optimal parameters through stochastic gradient descent

The aim of feedforward control is to find a parameterization $f(r) \approx P^{-1}r$ such that $e \approx 0$, see Figure 1. To this end, consider a parameterization of f that is linear in the parameters θ:

$$f(r, \theta) = \psi(r)^T \theta.$$

An example of this structure is that of mass feedforward for a 2x2 MIMO system with interaction, constructed as

$$\begin{bmatrix} f_m^1(r,\theta) \\ f_m^2(r,\theta) \end{bmatrix} = \begin{bmatrix} \ddot{r}^1 & \ddot{r}^2 & 0 & 0 \\ 0 & 0 & \ddot{r}^1 & \ddot{r}^2 \end{bmatrix} \begin{bmatrix} \theta^1 \\ \theta^2 \\ \theta^3 \\ \theta^4 \end{bmatrix}.$$

In order to minimize $\mathcal{J} = ||e||^2$, with

$$e = (I + PC)^{-1} r - (I + PC)^{-1} P\, f = Sr - Jf,$$

the gradient $g(\theta) = \frac{d\mathcal{J}}{d\theta}$ is computed as

$$g(\theta) = -2\psi(r) J^T e(\theta).$$

For SISO systems, a trick using adjoints enables the computation of $g(\theta)$ through a direct measurement on J^T, such that the optimal feedforward parameters can be found through gradient descent without model knowledge. However, this does not extend well to multivariable systems: generating gradients requires $n_i \times n_o$ experiments per iteration and is comparable to tuning by turning one knob at a time.

Instead of expensive exact gradients, an unbiased gradient estimate can be generated through one experiment regardless of the size of the MIMO system [1]. To this end all experiments are run simultaneously ('turn all knobs') in randomized directions, leading to an estimate $\hat{g}(\theta)$ of the gradient. This estimate is used in a stochastic gradient descent algorithm to update the parameters according to

$$\theta_{j+1} = \theta_j + \epsilon_j \hat{g}(\theta_j).$$

The optimal step size ϵ_j is determined through an additional experiment.

Experimental results
The approach is illustrated through two examples. The first is a simulation example using a 21x21 system, for which an unparameterized feedforward signal is optimized, i.e., $\psi = I$. The results in Figure 2 show that the stochastic gradient descent ILC approach using a randomized gradient estimate results in fast convergence compared to deterministic approaches that use the full gradient.

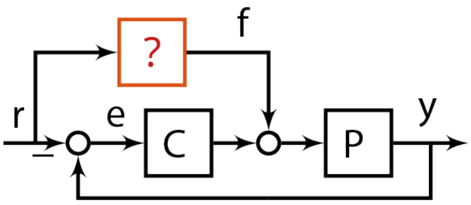

Figure 1 The aim of parameterized feedforward control is to find $f(r) \approx P^{-1}r$ such that $e \approx 0$.

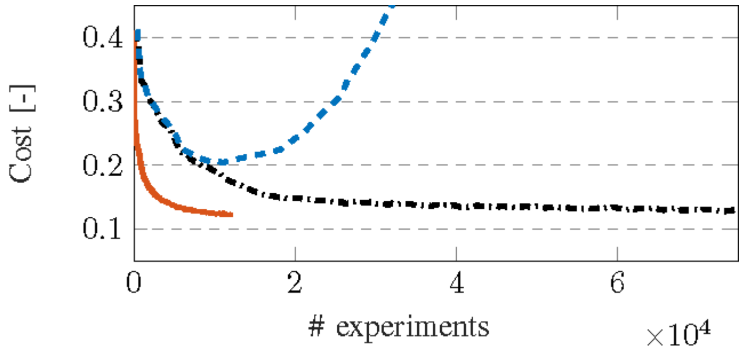

Figure 2 *Stochastic gradient descent ILC (red) is faster than deterministic approaches (black) that may diverge when data is noisy (blue).*

The approach is also applied in experiments to the parameterized feedforward control of an industrial flatbed printer, see Figure 3. The printer is a 3x3 system with inputs in two translation and one rotation direction. The position, velocity and acceleration feedforward parameters are tuned automatically in a small number of experiments, as shown in Figure 4.

Figure 3 Arizona industrial flatbed printer.

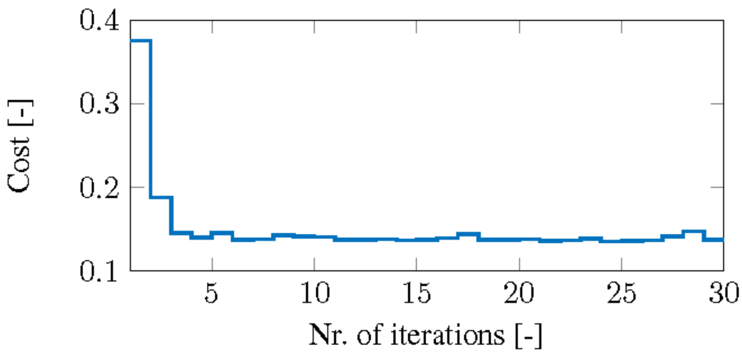

Figure 4 The automated model-free tuning of feedforward parameters for the flatbed printer requires only a small number of iterations.

2) Nonlinear filters in ILC: beating the trade-offs between iteration-varying (noise) and iteration-invariant (reference) disturbances [3]

Iterative learning control (ILC) aims to learn a feedforward signal f that fully compensates repeating disturbances through repeated experiments. While standard ILC can attenuate repeating disturbances completely, leading to high performance, it may also amplify iteration-varying disturbances up to a factor two. A standard ILC update of a feedforward signal f is given by

$$f_{j+1} = Q(f_j + \alpha L e_j),$$

with learning filter L an approximation of the system inverse, and robustness filter Q a lowpass filter. The learning gain $\alpha \in (0,1]$ can be used to reduce the amplification of iteration-varying disturbances, but this also reduces the learning speed significantly as illustrated in Figure 5. The aim of this research is to develop an approach that achieves both small converged errors and fast convergence.

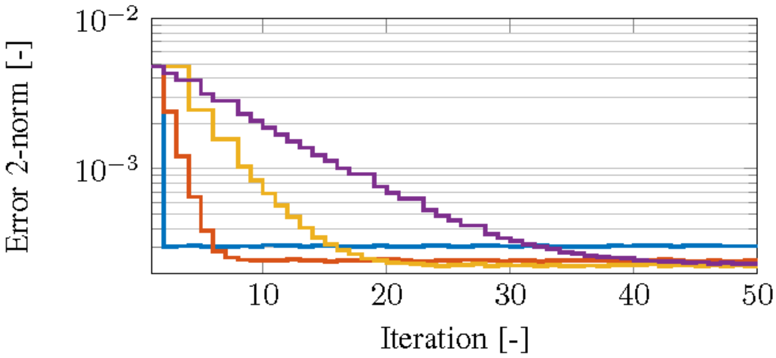

Figure 5 *ILC with learning gain α equal to 1(blue), 0.5 (red), 0.2 (yellow) and 0.1 (purple). Reducing α leads to lower errors because the noise amplification is reduced, at the cost of convergence speed.*

Approach: nonlinear iterative learning control

The main idea is to use a deadzone nonlinearity to differentiate between varying and repeating disturbances based on their amplitude characteristics and apply different learning actions: fast attenuation of repeating disturbances, and slow averaging of varying disturbances [3]. In Figure 6 the repeating and iteration-varying disturbances for 20 experiments are shown.

The deadzone nonlinearity with width δ and gain γ is included in the feedforward update as follows:

$$f_{j+1} = Q\left(f_j + \alpha L e_j + L\varphi(e_j)\right),$$

with

$$\varphi\left(e_j(k)\right) = \begin{cases} 0, & \text{if } |e_j(k)| \leq \delta \\ \left(\gamma - \dfrac{\gamma\delta}{|e_j(k)|}\right) e_j(k), & \text{if } |e_j(k)| > \delta. \end{cases}$$

The deadzone nonlinearity satisfies an incremental sector condition with γ, which enables convergence analysis for frequency-domain ILC, lifted ILC and the related technique of repetitive control. The resulting convergence conditions are reminiscent of the conditions employed in standard ILC. For example, frequency-domain nonlinear ILC is monotonically convergent in the vector 2-norm if

$$\left\| Q\left(1 - \alpha JL - \frac{\gamma}{2} JL\right)\right\|_{\mathcal{L}\infty} + \frac{\gamma}{2}\|QJL\|_{\mathcal{L}\infty},$$

a condition which can be evaluated easily using measured or identified frequency response data of the system.

Simulation results

Nonlinear frequency-domain ILC is validated in simulations. The results in Figure 7 show that nonlinear ILC achieves both fast convergence and low converged errors, removing the trade-off that is present in standard ILC.

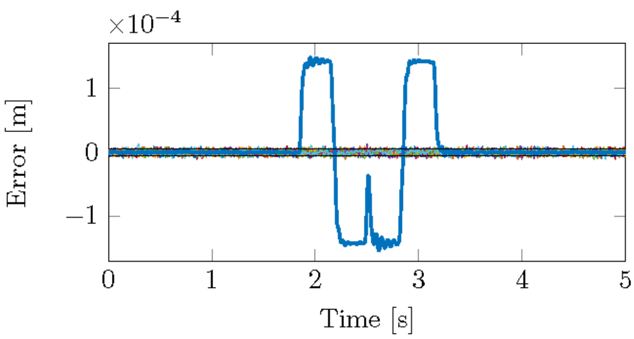

Figure 6 *Iteration-varying and repeating (blue) disturbances.*

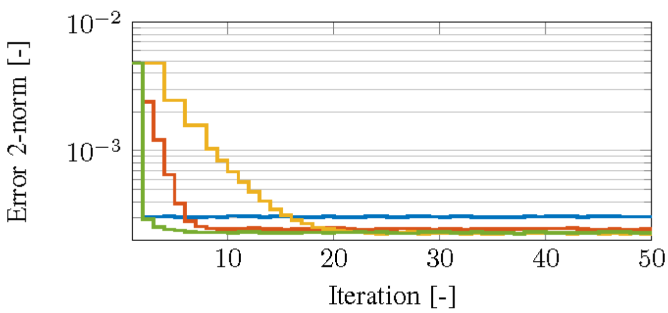

Figure 7 *Nonlinear ILC (green) achieves both fast convergence and small converged errors, removing the trade-off that is present in standard ILC.*

3) Neural networks for flexible feedforward: cost functions, model structures and training data

Neural networks are promising for flexible feedforward control, but combining them in a harmonious way with state-of-the-art feedback control is subtle and requires care. In particular:

- The cost function used for training should reflect the aim of minimizing the tracking error, as $||f_{train} - f_{nn}||$, with $e(f_{train}) = 0$, being small does not necessarily mean that $e(f_{nn})$ will be small.
- The model structure should allow for non-causal feedforward, as many systems contain delays.
- Training data, consisting of representative references and feedforward signals, should be generated in closed-loop, for example using ILC, as nonlinearities manifest along trajectories.

Experimental results on the carriage of the Arizona flatbed printer (Figure 3) for two types of non-causal neural networks are shown in Figure 8. The results show that good performance can be achieved. Recurrent neural networks are shown to be sensitive to overfitting, reducing the performance.

4) Weighting the errors that matter: cross-coupled iterative learning control

For contour tracking applications, the error in time domain is less important than the deviation from the contour. Cross-coupled ILC can be used to design feedforward signals for these specific cases, by using a cost function that weights this contour error explicitly. The cost function also weights the error tangential to the contour error, to allow for specifying different aims in different parts of the trajectory. For example, one might want to slow down in sharp corners and make up for lost time when moving straight.

Simulation results in Figure 9 show the improved tracking of the contour (black) when cross-coupled ILC is applied (blue, dashed) compared to the case without ILC (red, dotted).

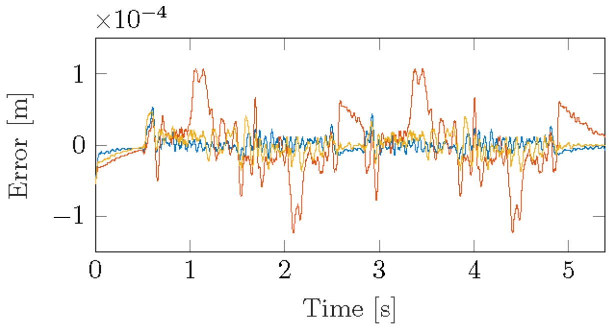

Figure 8 *Errors for a reference outside the training set resulting from f_{train} (blue), a non-causal time-delay neural network (yellow) and a non-causal recurrent neural network (red)*

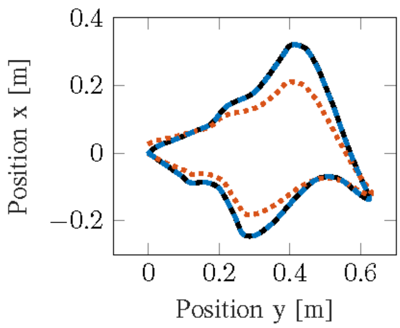

Figure 9 *Iteration-varying and repeating (blue) disturbances.*

Conclusion

Dedicated learning-based algorithms for feedforward control are developed by combining insights from precision mechatronics, control, and machine learning, leading to high performance in different use cases. To address the challenge of dealing with varying references, a method for efficient, automated model-free tuning of feedforward is parameters developed and the use of control-relevant neural networks for flexible feedforward is investigated. In addition, it is shown that a deadzone nonlinearity leads to reduced amplification of iteration-varying disturbances in ILC, and a dedicated ILC approach for contour tracking is developed.

Keywords: Feedforward Control, Iterative Learning Control, Nonlinear Control, Neural Networks

[1] Leontine Aarnoudse and Tom Oomen, 2022, "Automated MIMO motion feedforward control: Efficient learning through data-driven gradients via adjoint experiments and stochastic approximation", in: 2022 Modeling, Estimation, and Control Conference, Jersey City, New Jersey.

[2] T. Oomen and C. R. Rojas, 2017, "Sparse iterative learning control with application to a wafer stage: Achieving performance, resource efficiency, and task flexibility," in: Mechatronics, vol. 47, pp. 134–147.

[3] Leontine Aarnoudse, Alexey Pavlov, Tom Oomen, 2023, "Nonlinear iterative learning control: A frequency-domain approach for fast convergence and high accuracy", in: Proceedings of the IFAC 22st Triennial World Congress, Yokohama, Japan.