MACHINE LEARNING FOR INDUSTRIAL MOTION CONTROL

The recent explosive progress in machine-learning (ML) applications raises the question what ML has to offer for industrial motion-control systems. We aim to explore how the power of ML can be safely harnessed for real physical machines, and provide insight into design aspects that contribute to success. Successful ML-based applications at Sioux Technologies are presented, providing a hands-on perspective on the performance potential in motion-control applications.

LENNART BLANKEN AND CASPAR GRUIJTHUIJSEN

Introduction

The achievable performance of mechatronic positioning systems is facing limitations in the current state-of-art design paradigm. To meet future requirements on production speed, quality and cost, it is envisaged that a significant increase is required in the complexity of positioning systems. This leads to the manifestation of increasingly complex system behaviour. Examples include increasing numbers of motion axes, the presence of flexible dynamical behaviour in the control bandwidth and the associated coupling between axes, and increased susceptibility to disturbances from multiple physical domains such as friction, hysteresis, thermal effects, acoustics, noise from electronics, etc.

Mechatronic system design approaches typically focus on excellent electromechanical designs [1], which subsequently simplify control design. It is envisioned that this design paradigm will become infeasible, e.g., due to excessive cost of materials with favourable (thermo-)mechanical properties. Hence, the foreseen trend of increasing system complexity motivates to reconsider the holistic system design process, which ranges from sophisticated electromechanical designs to intelligent control and software solutions.

On another hand, explosive progress has been made in the field of machine learning (ML) over the past decades. Spurred by the availability of data and low-cost computation [2], this has led to astounding results in complex applications, for instance achieving superhuman performance in Go and Atari games [19]. Moreover, these results are achieved without any prior knowledge of the environment dynamics.

This raises the question what ML has to offer for high-tech positioning equipment, and in particular how exploitation of ML techniques can enable a revolution in dealing with the increasing system complexity during the holistic mechatronic system design process. Indeed, a huge performance potential seems readily attainable, as both data and computing power are abundantly available in high-tech mechatronic systems.

However, there is a fundamental difference between the presented examples of ML applications [19] and high-tech positioning systems: interaction with the physical world. Without interaction with physical systems, there need to be only mild requirements on the training process, i.e., required training time and convergence properties. Conversely, mechatronic applications, where interaction with the physical world is pivotal, have very strict requirements on the training process.
Most successful ML applications to mechatronic systems (for example, drone racing [5], tokamak plasma control [6], stratospheric balloon navigation [7], and bipedal robot soccer [8]) mitigate these challenges by training the ML algorithm in a simulation environment. However, the resulting control performance is directly determined by the quality of the system knowledge that is used to build the simulator. Especially in view of the foreseen increasing system complexity of high-tech positioning equipment, this excessive modelling burden is undesired.

The aim of this article is to explore opportunities and challenges associated with ML for high-tech motion-control applications. Successful adoption of ML imposes a unique set of requirements, since high-tech manufacturing machines are cyber-physical systems that interact with the real world, and machine downtime has a huge impact

AUTHORS' NOTE

Lennart Blanken is a mechatronic system engineer at Sioux Technologies in Eindhoven (NL), and part-time assistant professor at the Control Systems Technology group within the department of Mechanical Engineering at Eindhoven University of Technology. Caspar Gruijthuijsen is a mathware designer at Sioux Technologies.

This article is the result of fruitful collaborations with many colleagues in academia and industry. They are gratefully acknowledged, in particular: Arend-Jan Beltman, Sorin Bengea, Mark van den Broek, Elisa Dankers, Rik Dekker, Max van Haren, Daan Nieuwenhuizen, Tom Oomen, Chuck Steijlen and Lense Swaenen.

Financial support was provided by the ECSEL Joint Undertaking under grant agreement 101007311 (IMOCO4.E), which receives support from the European Union Horizon 2020 research and innovation programme.

lennart.blanken@sioux.eu
www.sioux.eu
www.tue.nl/cst

on their economic value. In [3], it is argued that the learning on the physical system:

- shall be fast, since dedicated training experiments lead to production loss and fast adaptation is desirable in case of varying operating conditions, such as temperature changes due to internal dissipation and environmental variations;
- and shall be safe, since damage to the machine is in general unacceptable.

In the next section, we outline our view on the implications of these requirements on appropriate design of ML-based approaches for control. In the subsequent section, we present a range of suitably designed solution approaches that vary in their use of prior knowledge, ranging from data-enhanced yet dominantly physics-motivated control designs to more black-box-oriented ML controllers. Each technique is illustrated through a case study of an application at Sioux Technologies.

Machine learning for motion control

Machine learning is not so easy to apply in industrial practice of motion control due to the many decisions that need to be made, and since the impact of these decisions on the posed motion-control requirements is often unclear: what controller architecture, structure and complexity should be chosen, how the controller should be trained, how learning efficiency and safety can be guaranteed, etc.

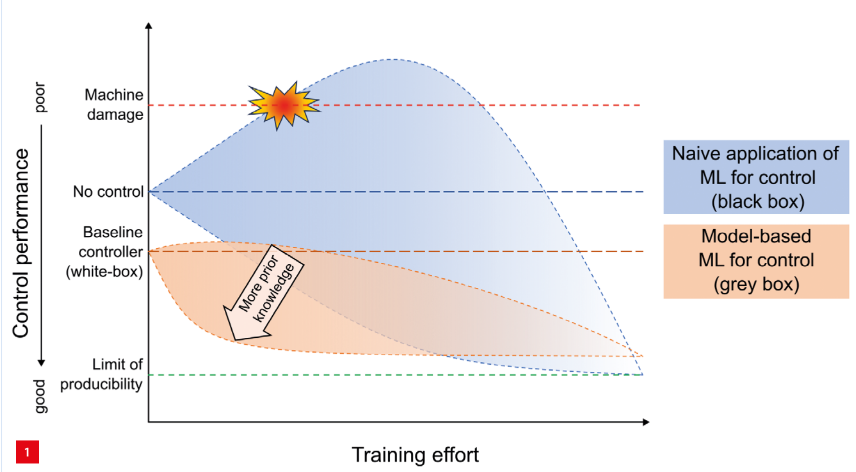
For example, neural networks are extensively in use as black-box function approximators in ML, but to control engineers they are unsatisfactory due to the difficulty in interpretation and lack of guarantees on learning speed and safety. In this section, we present insight into several aspects that contribute to successful applications of ML to motion control.

Role of prior knowledge

The central hypothesis in this article is that the judicious use of prior knowledge is pivotal for successful application of ML in motion control, as it can accelerate and safeguard the learning process. This is illustrated by Figure 1:

- *Black-box:* ML approaches do not use prior knowledge and typically require excessive training effort, i.e., the learning is slow. Additionally, the stability of the learning process is difficult to guarantee, i.e., the learning is not guaranteed to be safe.
- *Grey-box:* model-based ML approaches make explicit use of both data and prior knowledge and can lead to faster learning with guarantees on stability and performance.

In the next paragraphs, we discuss in what ways prior knowledge can be embedded in ML for motion control. In particular, the following essential elements are discussed: the selected controller structure, the used learning algorithm, and the data used for training.



The central statement in this article is that the judicious use of prior knowledge (e.g. models) in machine-learning algorithms is fundamental for successful applications in motion control, as it can accelerate and safeguard the learning process. Naive application of ML techniques to motion control often leads to unsatisfactory results, including machine damage.

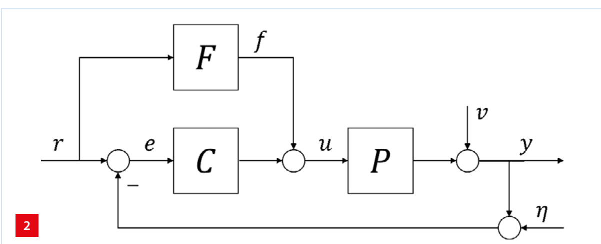
Control structure

Physics-based information can be directly employed in the control structure. The main point here is that domain-specific engineering knowledge that has proven itself during the last decades should be appropriately retained. Consider the control architecture shown in Figure 2 that is typically used in the motion-control domain. Here, P is the system to be controlled, and C and F are the feedback and feed-forward controllers. The servo error e can be represented as:

$$e = S(I - PF)r - Sv - S\eta$$

This equation features sensitivity function $S = (I + PC)^{-1}$, motion reference signal r, and actuator and measurement disturbance signals v and η, respectively. Since the goal of feedback control is to attenuate disturbances v and η, feedback controllers are often tuned based on characterisations (models) of these disturbances. Feedforward controllers are typically designed based on knowledge of the inverse system dynamics, since the reference r is perfectly tracked if $F = P^{-1}$. Industrial practice is to design the feedforward controller as a low-order physics-based approximation of the inverse system dynamics. Using Newton's second law, for instance of the form:

$$m\frac{\mathrm{d}^2y}{\mathrm{d}t^2} + b\frac{\mathrm{d}y}{\mathrm{d}t} + ky = u$$



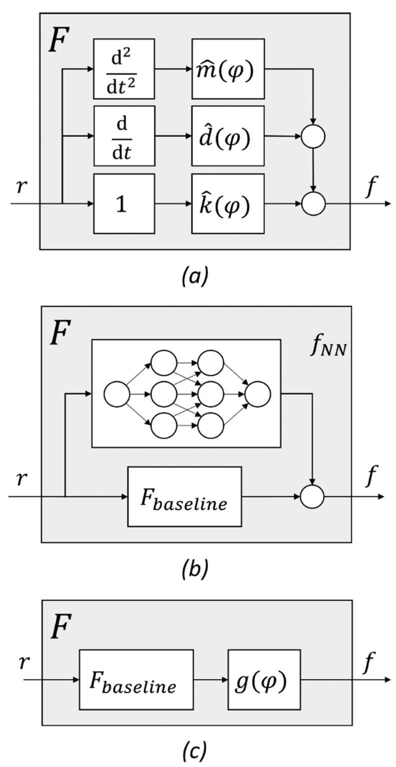
Typical motion-control architecture.

Here, m, b, k represent the mass of the system, the damping and the stiffness to the fixed world, respectively. This physics-based model directly motivates the feedforward design that is typically preferred by motion-control engineers:

$$f = \hat{m}\frac{\mathrm{d}^2 r}{\mathrm{d}t^2} + \hat{b}\frac{\mathrm{d}r}{\mathrm{d}t} + \hat{k}r$$

Here, $\mathrm{d}^2r/\mathrm{d}t^2$ and $\mathrm{d}r/\mathrm{d}t$ are the acceleration and velocity profiles of the setpoint r, respectively. The parameters $\hat{m}$, $\hat{b}$, $\hat{k}$ are to be tuned, as physical quantities m, b, k are not exactly known. Aided by their physical interpretability and the simple control structure, in traditional motion control the parameter tuning is often performed manually [4].

Given that this physics-based approach to controller structure has proven itself over the past decades, this domain-specific expertise should not be discarded in favour of full black-box control structures. In the selection of ML-based control structures for increasingly complex systems, we can build upon this expertise. In particular, low-order physics-based control structures can be enriched through ML.
Examples are shown in Figure 3. In Figure 3a, physically interpretable feedforward gains are modelled as high-complexity functions of a generalised variable φ, see also [11]. The add-on neural network in Figure 3b only needs to account for what is not compensated by the baseline controller, see also [14, 17]. In Figure 3c, the control gain is scaled by a high-complexity function, e.g., to compensate for effects such as cogging.



3

Examples of ML-enriched control structures.

Learning algorithm
Prior knowledge can be used in the selected learning algorithm to ensure fast and safe learning in multiple ways. The optimisation criteria for learning can be designed in a control-relevant manner, that is to explicitly reflect the control goal of minimising the servo error. This involves suitable weighting of typical optimisation criteria with dynamical models of the closed-loop system [14]. Also, by embedding models in the optimisation algorithms, i.e., the solutions to the posed optimisation problems, the learning process can be accelerated and safeguarded [3, 9, 11]. In addition, prior knowledge can be exploited to explicitly impose certain properties on the learned controller, e.g. via constrained learning [16] and model-based regularisation [11, 15].

Training data and process
When working with real machines, the need for dedicated training time on the physical system should in general be avoided, e.g., to avoid costly production losses in manufacturing machines. This can be approached in several ways.

When opting for learning on the physical system, fast learning with guaranteed stability is essential. This can be achieved by limiting the required training exploration. For instance, prior knowledge can be exploited to suitably restrict the control design space via judicious selection of the controller structure and learning algorithm. Also, the learning can be targeted on the operational use-cases only, e.g., the repeated execution of similar tasks in motion systems can be exploited directly in iterative learning approaches; see also the forthcoming Case study 1.

Training can also be performed based on synthetic data. By adopting high-fidelity simulators, ML-based controllers can be trained in inherently safe environments and in accelerated (simulation) time. The goal is to (pre)train the controller until it is sufficiently good for transfer to the physical system. It can then be applied as is (*zero shot*), or further finetuned using only a few experiments. The main

Table 1
Overview of case studies and their use of prior knowledge.

	Case study 1	Case study 2	Case study 3
Controller structure	Physics-based parametrisation (white-box)	Physics-based parametrisation enriched with ML-functions (grey-box)	Neural network (black-box) as add-on to physics-based baseline (white-box)
Learning algorithm	Joint model- and data-based iterative learning	Joint model- and data-based iterative learning	Residual reinforcement learning
Training data	Operational data from physical system	Operational data from physical system	Simulation environment based on model

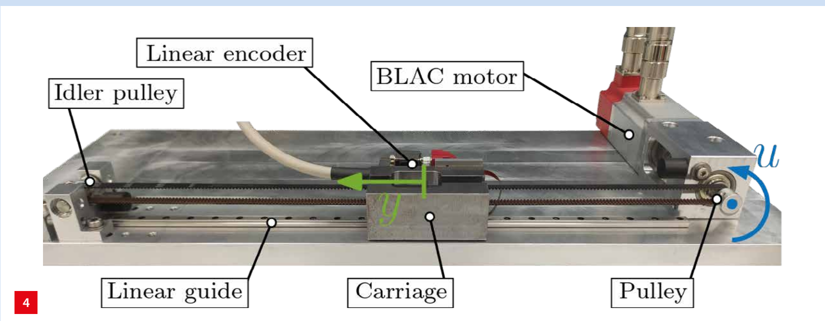
Case-study system: belt-driven stage

The belt-drive system in Figure 4 is a prime example of a system where advanced motion-control solutions are required to achieve high motion performance, which makes it a suitable case-study system. In fact, its deliberate cost-effective hardware design directly limits the performance of traditional control designs. Relevant application areas include industrial 3D-printing systems.

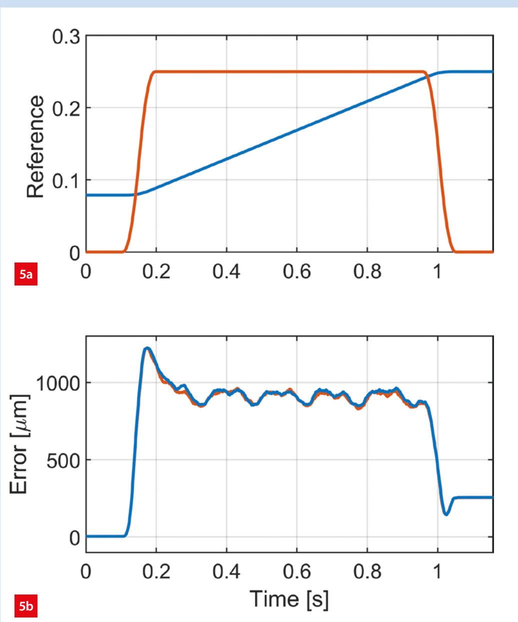
The control goal is accurate tracking of a motion trajectory r [m] with the carriage position y [m] by means of input force u [N] applied to the belt drivetrain. The motion trajectory and tracking error $e = r - y$ using a baseline PID feedback controller are shown in Figure 5. It can be observed that:

- the tracking error correlates with the acceleration and velocity profiles of the setpoint, which motivates the use of acceleration and friction feedforwards;
- however, during the constant-velocity phase of the setpoint, a dominant ripple in the tracking error can be observed. This ripple reproduces over experiments, yet cannot be compensated for using standard mass-damping-stiffness feedforwards.

The control goal for the case studies in this section is to mitigate this tracking error, where different ML-based approaches are taken to compensate for the observed ripple.



Belt-drive system.



Tracking of motion trajectory for the case-study system.
(a) Position profile (blue) [m] and velocity profile (red) [m/s] of the scanning reference.
(b) Tracking error in two experiments (blue and red) using a baseline controller.

challenge herein is the unmodelled gap between the simulation environment and physical reality. To overcome this so-called *sim2real* gap, it is essential that high-quality prior knowledge is used to construct the simulation environment. In addition, there are methods to robustify the pretrained controller to unmodelled system behaviour, e.g., via domain randomisation [18].

Case studies

In this section, we present several examples of solution directions for successful ML-based motion control. Each presented approach employs different levels of prior knowledge encoded through the used control structure, learning algorithm and training process, which is summarised in Table 1. The efficacy of each approach, including learning speed and safety, is illustrated through an application to the case-study system at Sioux Technologies.

Learning of physics-based feedforward parameters

The abundance of measurement data in mechatronic systems can be exploited to improve the tuning of the physics-based control structures that are often preferred in traditional motion control. Joint model- and data-based iterative learning [3, 9] provides automated methods for fast, safe and accurate parameter tuning of fixed-structure controllers. Prior knowledge in the form of dynamical models plays an essential role to accelerate the learning process and guarantee safety. This is illustrated next.

A fixed-structure controller is employed of the following general form:

$$f = \sum_{i=1}^{n_\theta} \theta_i \psi_i = \Psi\Theta$$

Here, $\Theta = [\theta_1, \theta_2, \dots, \theta_{n_\theta}]^\top$ are the parameters to be tuned, and $\Psi = [\psi_1, \psi_2, \dots, \psi_{n_\theta}]$ are so-called basis functions. A typical motion feedforward controller is recovered by choosing $\psi_1 = \mathrm{d}^2r/\mathrm{d}t^2$ (mass feedforward), $\psi_2 = \mathrm{d}r/\mathrm{d}t$ (damping feedforward) and $\psi_3 = r$ (stiffness feedforward). In model-based iterative learning [3, 9], the parameters Θ are iteratively optimised with guarantees on the learning speed and safety. The concept is as follows. Let the subscript j indicate the experiment index. After each experiment j, the parameters for the next experiment $j + 1$ are updated through the following control-oriented optimisation problem:

$$\begin{aligned}\Theta_{j+1} &= \arg\min_{\Theta_{j+1}} \left\| \hat{e}_{j+1}(\Theta_{j+1}) \right\|_2 \\ &= \arg\min_{\Theta_{j+1}} \left\| e_j - \widehat{SP}\Psi(\Theta_{j+1} - \Theta_j) \right\|_2\end{aligned}$$

Here, $\hat{e}_{j+1}(\Theta_{j+1})$ is a prediction of the tracking error in the next experiment. The crucial aspect is that fast and safe learning can be achieved through the combined model- and data-based prediction:

Case study 1: Safe and fast learning of a fixed-structure feedforward

A fixed-structure feedforward controller is learned for the belt-drive system using the presented approach for data-based tuning. Motivated by physics, the feedforward consists of Coulomb friction, viscous friction and mass feedforward contributions:

$$f = \theta_a \frac{\mathrm{d}^2 r}{\mathrm{d}t^2} + \theta_v \frac{\mathrm{d}r}{\mathrm{d}t} + \theta_c \operatorname{sign}\left(\frac{\mathrm{d}r}{\mathrm{d}t}\right)$$

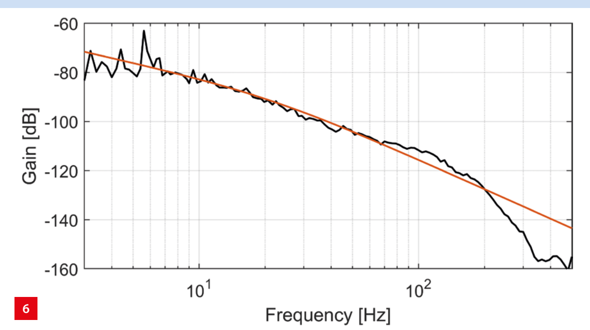
The corresponding parameters θ_c [N], θ_v [Ns/m] and θ_a [kg] are iteratively learned using operational data and an approximate model of the system; see Figure 6. The model contains significant modelling errors, as both Coulomb friction and the high-frequent decoupling of the carriage are not modelled, yet it will be shown that it is of sufficient quality to guarantee stable learning. The operational data is generated by repeated execution of a typical scanning setpoint.

The results in Figures 7 and 8 illustrate the efficacy of the model-based learning:

- fast learning is achieved: a handful of experiments is needed to convergence, comprising a total of only 10 seconds of measurement time;
- the learning process is stable, i.e., safe: the tracking error is reduced consistently from experiment to experiment, from 0.91 mm RMS error in experiment 1 to 38 µm in experiment 10.

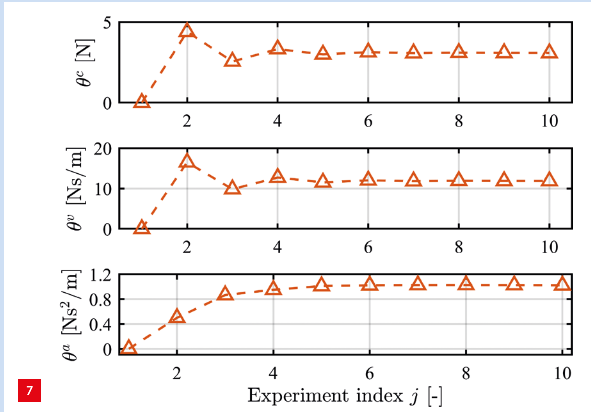
Also, the performance limit of the fixed-structure controller becomes apparent:

- a dominant periodic tracking error remains that reproduces over the experiments; this cannot be compensated by the existing feedforward structure since it manifests during the constant-velocity phase of the setpoint, where the feedforward force is constant.



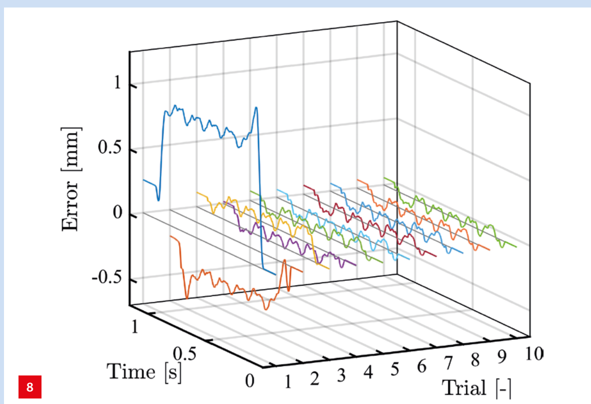
6

Bode magnitude plot of the true system (black; FRF measurement) and the approximate model (red) that is of sufficient quality to guarantee fast and safe learning.



7

Learning of physics-based feedforward parameters: stability and fast learning is achieved through the use of model knowledge.



8

Performance limit of a fixed-structure controller: through fast and safe tuning the tracking error decreases asymptotically, but a residual error remains that cannot be compensated with the implemented control structure.

- the use of approximate model $\widehat{SP}$ of the real system SP guarantees that the learning converges, i.e., $||e_{j+1}|| \leq \gamma ||e_j||$, where the learning rate γ depends on the model quality; in fact, it can be shown that $\gamma < 1$ for modelling errors up to 100% [3].
- the use of past operational data e_j ensures that high performance is achieved: if $\gamma < 1$, then $||e_j||$ exponentially goes to zero as j increases.

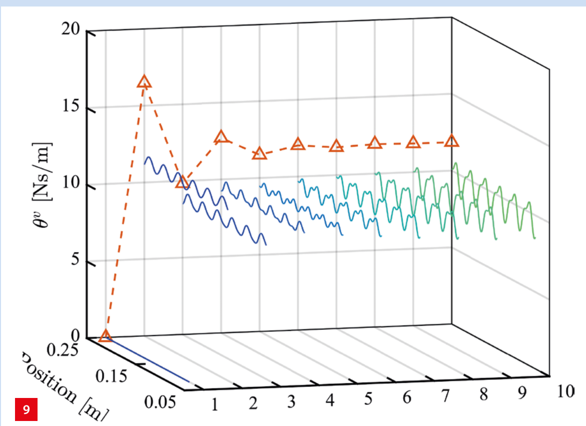
Case study 1 illustrates the guaranteed fast and safe learning behaviour through the use of prior knowledge, and the inherent performance limitation associated with the fixed controller structure.

Physics-motivated control structures work well if the physics that govern the system are well understood: this is traditionally exploited in motion control. However, application of this control design methodology to increasingly complex systems leads to excessive burdens

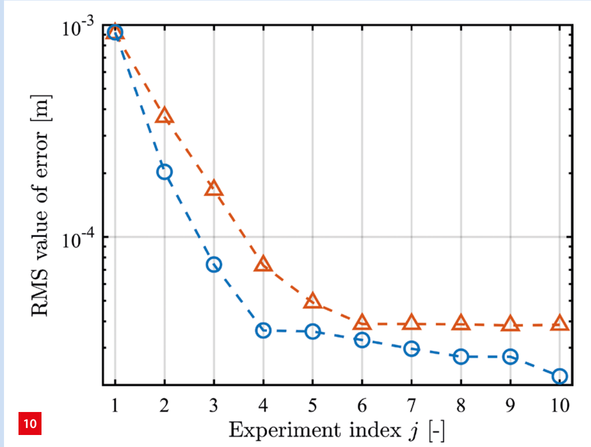
Case study 2: Machine learning of a position-dependent feedforward

Case study 1 has illustrated the performance limitation of fixed-structure feedforward controllers: after learning, the remaining residual error could not be compensated by the physics-motivated mass-damping-friction feedforward structure. However, the observed realisation of the residual error gives direction to how the controller structure can be enriched: a dominantly periodic signal is observed during the constant-velocity phase of the setpoint, where the period correlates with the drive-pulley circumference. This knowledge is exploited to suitably enrich the controller structure using Gaussian Process (GP) regression [13, Chapter 2].

In summary, a GP is a generalisation of a Gaussian probability distribution: whereas probability distributions describe (scalar) random variables, GPs describe random functions. Such function distributions can for example represent the position-dependent control parameter. When using GPs in a framework called Bayesian inference, they can be used to learn unknown functions from data. And crucially, certain high-level properties of the function to be learned can be enforced, while still allowing for sufficient modelling freedom. This is what is done next: the periodic behaviour of the residual error can be appropriately included in the feedforward.



9 ML-enhanced learning of the damping feedforward function (blue to green). The position-invariant result of Case study 1 is included for reference (red).



10 RMS value of the tracking error of the ML-enhanced feedforward controller (blue), and results of Case study 1 (red).

The physics-based feedforward structure is enriched using machine learning:

$$f = \theta_a \frac{\mathrm{d}^2 r}{\mathrm{d}t^2} + \boldsymbol{\theta}_v(\boldsymbol{x}) \frac{\mathrm{d}r}{\mathrm{d}t} + \theta_c \operatorname{sign}\left(\frac{\mathrm{d}r}{\mathrm{d}t}\right)$$

Here, the viscous friction feedforward gain is now a function $\theta_v(x)$ [Ns/m] of the drive-pulley position x. Using GP-based iterative learning [11], the feedforward gains [θ_a, $\theta_v(x)$, θ_c] are learned over experiments. Here, the unknown function $\theta_v(x)$ is modelled as a GP, where the GP encodes the available prior knowledge: $\theta_v(x)$ is restricted to the class of functions that is periodic with $x = 2\pi$ [rad] and has a specified level of smoothness. This class of functions is still rich enough to successfully model the complex friction behaviour of the belt-drive system.

The results in Figures 9 and 10 demonstrate that through GP-enhanced learning of the physics-based controller:

- a position-periodic viscous friction feedforward is successfully learned in only a few experiments, i.e., the learning is fast;
- the learning process is stable, i.e., safe, and the learned feedforward function remains close to physical expectations enforced through the GP;
- a performance improvement is achieved of almost a factor 2 in terms of RMS tracking error as compared to the results of Case study 1 (22 μm versus 38 μm).

on the physical modelling. Without high-fidelity physical understanding, it is nontrivial how to expand controller structures and how to tune them.

Learning of ML-enriched feedforward functions

A systematic solution direction to improve performance is to enrich the controller complexity through ML, yet retain the physics-motivated structure that has proven itself in the field of industrial motion control. For example, the fixed-structure feedforward controller can be augmented as follows, see also Figure 3a:

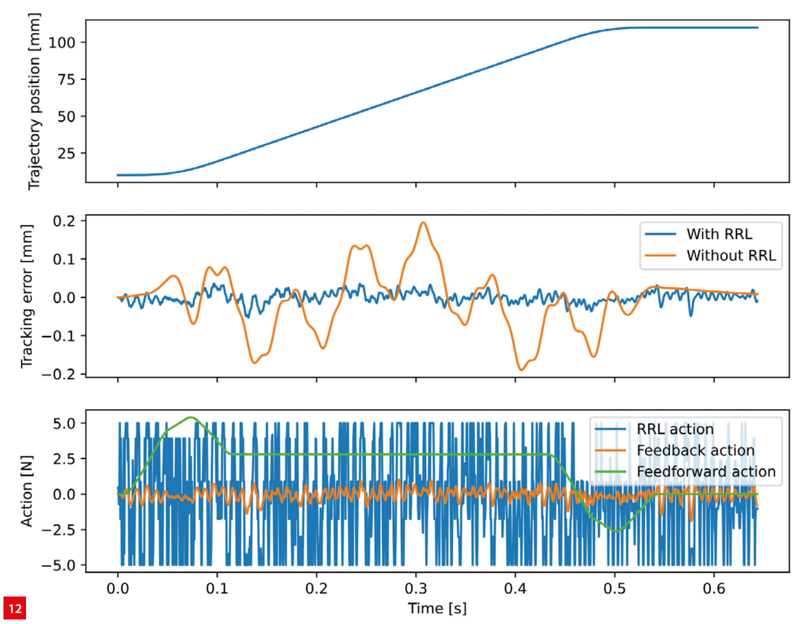
$$f = \sum_{i=1}^{n_\theta} \theta_i(\varphi)\psi_i$$

Here, the parameters are now functions of some variables φ. Enabled by ML techniques, these functions can be of

Case study 3: Residual reinforcement learning

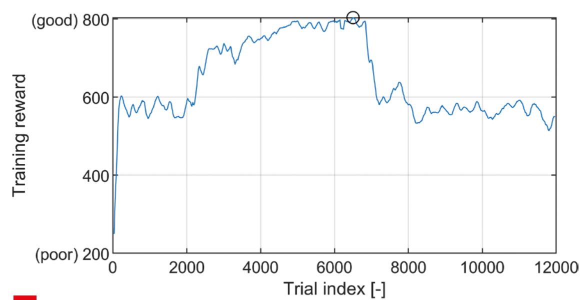
This case study resumes from the endpoint of Case study 1: good tracking behaviour was achieved on the belt-drive system, but a residual error remained. Where Case study 2 addressed this residual by enriching the controller complexity with ML components in a physics-guided way, Case study 3 considers the residual dynamics as entirely black-box, addressing them via residual reinforcement learning (RRL) [20].

Because of the explorative character of this RRL case study, a high-fidelity simulator is used that includes the complex position-dependent dynamics of the belt drivetrain. These dynamics are assumed to be entirely unknown to the RRL. The RRL policy is trained to compensate for any remaining tracking error that the base controller (the controller from Case study 1) does not cover, see Figure 3b for the used control structure.



12

Tracking results of RRL applied to the belt-drive system.



13

Training behaviour of RRL, where the best control performance (highest training reward) is achieved after more than 6,000 trials.

The results in Figures 12 and 13 illustrate that:

- RRL enables significant reduction of the tracking error compared to the baseline controller, i.e., effectively compensates for the complex position-dependent drivetrain dynamics.
- The learned control action is highly erratic, although it effectively compensates the realisation of the residual error. This is attributed to the black-box structure of the controller that does not make use of prior knowledge. Note that the neural network output is saturated at ±5 N to facilitate some level of machine safety.
- More than 6,000 trials are needed to learn the optimal controller, i.e., the learning process is much slower than in Case studies 1 and 2. It is noted that since the learning takes place in accelerated simulation time, this is not necessarily an issue. Note that around 7,000 trials an abrupt performance degradation occurs, known as *forgetting* during ML training.

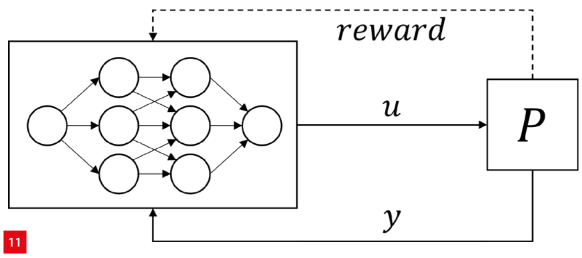
Considering the transfer of the trained controller to the physical system (not included in this case study) it may be clear that, although extreme performance is realised in simulation, the observed erratic control action and relatively slow learning without guaranteed stability remain important hurdles.

extreme complexity to compensate for intricate dynamics of which the particular physical realisations are not a priori known. Examples include complex friction behaviour, and position-dependent and nonlinear dynamics.

A particularly suitable ML methodology for motion-control applications is Gaussian Process (GP) regression. The interested reader is referred to [13, Chapter 2] for an excellent introduction into GPs, and [10, 12] for examples of GP applications in motion control. The key point is that GPs enable to incorporate engineering knowledge on the unknown function that is to be learned. Case study 2 presents a successful application that learns an intricate friction feedforward term.

Learning of black-box residual feedforwards

From Case study 2, it is clear that ML techniques, encapsulated in the safe confines of a physics-motivated control structure, can lead to significant performance improvements. However, with increasing complexity of mechatronic systems, physics-motivated fixed control structures might not allow to adequately capture all relevant dynamics, for instance if these are not well understood or hard to model. A black-box ML approach can be attempted to capture such dynamics.



11

Black-box ML techniques learn environment dynamics through rewards for explorative interactions.

Reinforcement learning (RL) is a black-box ML approach that learns a control law (policies) through suitably rewarded interaction with an environment that is assumed to be unknown, see Figure 11. Because this interaction is mostly explorative in the early phase of training, it is time consuming and potentially unsafe for real physical systems. This hampers the applicability of RL in mechatronics.

Residual RL (RRL) is an approach that aims to address the potentially unsafe behaviour of RL [16, 20]. RRL relies on a pre-tuned white-box base controller to deliver reasonable performance and warrant machine safety. In addition, a black-box RRL policy is superposed that complements the control action, see Figure 3b. The base controller accelerates and guides the RRL training. It also enhances safety, as the added control action is typically small. Yet, the dependency on a high-fidelity simulator and the associated *sim2real* gap remains an obstacle for effective and practical usage of RRL.

Case study 3 explores RRL for the belt-drive system. It shows that this highly data-driven technique has the potential to surpass the results from more physics-based techniques (e.g., Case studies 1 and 2), yet this potential comes at the price of learning speed and safety, and is based on the (potentially unrealistic) assumption of a high-fidelity simulator.

Conclusion

The use of machine-learning techniques in motion control looks highly promising, and could revolutionise the creation process of industrial mechatronic systems. This article has presented a perspective on how the power of ML can be safely harnessed for real physical machines. The central statement is that the judicious use of prior system knowledge is fundamental for successful implementations, as it can accelerate and safeguard the learning processes. This prior knowledge can be embedded in the controller structure, the learning algorithm and the training process. ML-based motion-control concepts have been presented that vary in their use of ML algorithms and prior knowledge, and their efficacy has been illustrated on a case-study system at Sioux Technologies.

REFERENCES

[1] R. Munnig Schmidt, G. Schitter, and J. van Eijk, *The design of high performance mechatronics*, 1st edition, Delft University Press, Delft (NL), 2011.
[2] M.I. Jordan, and T.M. Mitchell, "Machine learning: Trends, perspectives, and prospects", *Science*, vol. 349, pp. 255-260, 2015.
[3] T. Oomen, "Learning in machines", *Mikroniek*, vol. 58 (6), pp. 5-11, 2018.
[4] T. Oomen, "Control for precision mechatronics", *Encyclopedia Syst. Control*, pp. 267-276, Springer International Publishing, 2021.
[5] E. Kaufmann, *et al.*, "Champion-level drone racing using deep reinforcement learning", *Nature*, vol. 620, pp. 982-987, 2023.
[6] J. Degrave, *et al.*, "Magnetic control of tokamak plasmas through deep reinforcement learning", *Nature*, vol. 602, pp. 414-419, 2022.
[7] M.G. Bellemare, *et al.*, "Autonomous navigation of stratospheric balloons using reinforcement learning", *Nature*, vol. 588, pp. 77-82, 2020.
[8] T. Haarnoja, *et al.*, "Learning agile soccer skills for a bipedal robot with deep reinforcement learning", *Science Robotics*, vol. 9 (89), 2024.
[9] L. Blanken, *et al.*, "Batch-to-batch rational feedforward control: From iterative learning to identification approaches, with application to a wafer stage", *IEEE/ASME Trans. Mechatronics*, vol. 22 (2), pp. 826-837, 2017.
[10] M. Poot, *et al.*, "Gaussian processes for advanced motion control, *IEEJ J. Ind. Appl.*, vol. 11 (3), pp. 396-407, 2022.
[11] M. van Haren, L. Blanken, and T. Oomen, "Iterative data-driven fixed-structure kernelized LPV Feedforward for motion systems", *in preparation for journal submission*, 2024.
[12] N. Mooren, G. Witvoet, and T. Oomen, "Gaussian process repetitive control with application to an industrial substrate carrier system with spatial disturbances", *IEEE Trans. Control Syst. Technol.*, vol. 31(1), pp. 344-358, 2023.
[13] C. Williams, and C.E. Rasmussen, *Gaussian processes for machine learning*. MIT Press, Cambridge (MA, USA), 2006.
[14] L. Aarnoudse, et al., Control-relevant neural networks for feedforward control with preview: Applied to an industrial flatbed printer", *IFAC J. Syst. Control*, vol. 27, 100241, 2024.
[15] L. Blanken, and T. Oomen, "Kernel-based identification of non-causal systems with application to inverse model control", *Automatica*, vol. 114, 108830, 2020.
[16] T. Staessens, T. Lefebvre, and G. Crevecoeur, "Adaptive Control of a Mechatronic System Using Constrained Residual Reinforcement Learning", *IEEE Trans. Ind. El.*, vol. 69 (10), pp. 10447-10456, 2022.
[17] J. Kon, et al., "Learning for precision motion of an interventional X-ray system: Add-on physics-guided neural network feedforward control", *Proc. IFAC 22st Triennial World Congress*, pp. 8126-8131, Yokohama (Japan), 2023.
[18] J. Moos, *et al.*, "Robust Reinforcement Learning: A Review of Foundations and Recent Advances", *Machine Learning and Knowledge Extraction*, vol. 4 (1), pp. 276-315, 2022.
[19] J. Schrittwieser, *et al.*, "Mastering Atari, Go, chess and shogi by planning with a learned model", *Nature*, vol. 588, pp. 604-609, 2020.
[20] T. Johannink, *et al.*, "Residual reinforcement learning for robot control", *Proc. IEEE Int'l Conf. Robotics Automation (ICRA)*, pp. 6023-6029, Montreal (Canada), 2019.