Applications

Christian Hespe*, Jan Kaiser, Jannis Lübsen, Frank Mayet, Matthias Scholz and Annika Eichler

Data-driven feedback optimization for particle accelerator application

Datengestützte Feedbackoptimierung für die Anwendung an Teilchenbeschleunigern

https://doi.org/10.1515/auto-2024-0170
Received December 5, 2024; accepted March 4, 2025

Abstract: For many engineering problems involving control systems, finding a good working point for steady-state operation is crucial. Therefore, this paper presents an application of steady-state optimization with feedback on particle accelerators, specifically the European X-ray free-electron laser. In simulation studies, we demonstrate that feedback optimization is able to reach a near-optimal steady-state operation in the presence of uncertainties, even without relying on *a priori* known model information but purely data-driven through input-output measurements. Additionally, we discuss the importance of including second-order information in the optimization to ensure a satisfactory convergence speed and propose an approximated Hessian representation for problems without second-order knowledge on the plant.

Keywords: steady-state control; feedback optimization; recursive least-squares estimation; particle accelerators

Zusammenfassung: Für viele regelungstechnische Probleme ist es wichtig, einen guten Arbeitspunkt für die Betriebsführung zu finden. Aus diesem Grund präsentiert dieser Artikel eine Anwendung von Ruhelagenoptimierung im geschlossenen Regelkreis auf Teilchenbeschleuniger, insbesondere den European XFEL. In Simulation wird demonstriert, dass Feedbackoptimierung fast optimale Ruhelagen für den Betrieb erreicht – sogar dann, wenn kein vorhandenes Modellwissen eingesetzt sondern datengestützt anhand von Eingangs- und Ausgangsmessungen gearbeitet wird. Zusätzlich wird diskutiert, dass Modellinformationen zweiter Ordnung entscheidend für die Konvergenzgeschwindigkeit der Optimierungsprobleme sind, und eine Näherung der Hesse-Matrix vorgeschlagen, die sich datengestützt berechnen lässt.

Schlagwörter: Ruhelagenregelung; Feedbackoptimierung; rekursive Methode der kleinsten Quadrate; Teilchenbeschleuniger

1 Introduction

For many business and engineering scenarios, formulating the task at hand as a mathematical optimization problem is a powerful and convenient approach. For example, many logistics and allocation problems can be solved using linear programming [1], and optimal control is an established tool in many engineering disciplines, both in its classical [2] and predictive form [3].

The current paper falls into the context of a particular engineering problem, the control of particle accelerators. As large and complex machines, accelerators present significant challenges in terms of the required performance and scale of the control problem. Specifically, this paper deals with steady-state optimization of particle beam properties. In this kind of problem, it is sought to find an input-output pair that keeps a physical plant at steady state while minimizing a user-defined cost function and is potentially subject to constraints. Most often, such problems arise as a high-level control task for plants that are either inherently stable by themselves or stabilized by a low level controller.

***Corresponding author: Christian Hespe**, Deutsches Elektronen-Synchrotron DESY, Notkestraße 85, 22607 Hamburg, Germany, E-mail: christian.hespe@desy.de
Jan Kaiser, **Frank Mayet and Matthias Scholz**, Deutsches Elektronen-Synchrotron DESY, Notkestraße 85, 22607 Hamburg, Germany, E-mail: jan.kaiser@desy.de (J. Kaiser), frank.mayet@desy.de (F. Mayet), matthias.scholz@desy.de (M. Scholz)
Jannis Lübsen, Institute of Control Systems, Hamburg University of Technology, Harburger Schloßstraße 22a, 21079 Hamburg, Germany, E-mail: jannis.luebsen@tuhh.de
Annika Eichler, Deutsches Elektronen-Synchrotron DESY, Notkestraße 85, 22607 Hamburg, Germany; and Institute of Control Systems, Hamburg University of Technology, Harburger Schloßstraße 22a, 21079 Hamburg, Germany, E-mail: annika.eichler@tuhh.de

Approached by means of classical optimization methods, we observe a key conflict: On the one hand, running classical model-based optimization, e.g., gradient descent or quasi-Newton schemes like BFGS [1], allows to take advantage of available information but is susceptible to uncertainties since no information about the actual plant is used. On the other hand, methods such as the Nelder-Mead algorithm [1] or Bayesian optimization [4], [5] compute solutions based on input-output measurements of the plant but can only partially incorporate model information to improve their performance. For particle accelerator control, Bayesian optimization [5], the Nelder-Mead simplex algorithm [6], and reinforcement learning [7] have proved successful when tuning machine parameters.

In the context of control, an essential component in improving robustness against uncertainties is feedback. In addition to a feedforward signal computed directly from the model, a feedback signal is determined from plant measurements to correct for deviations from the desired response. This idea is taken up upon by the method of *feedback optimization* [8], [9]. Instead of optimizing based only on an available model, the algorithm evaluates the cost function gradient at the measured plant output, closing a feedback loop around the optimizer and plant. Similar to integral behavior in linear time-invariant control loops [10], this structure allows to suppress uncertainties while maintaining optimal steady-state operation [11], [12] and track successfully in time-varying optimization problems [13].

As such, feedback optimization has attracted significant interest in the past, resulting in theoretical convergence guarantees [14]–[16] and diverse practical applications. For example, the method has been applied in the process industry [17], [18] for tracking optimal references, for congestion control in communication networks [19], and control of electric power grids [20], [21]. Furthermore, recent work has shown how feedback optimization can be applied in a data-driven and model-free scenario [22]–[24]. The idea of using online feedback in optimization has also been employed in model predictive control [3] and especially variants that employ the real-time iteration scheme [25]. However, while predictive control is aimed at dynamic control problems, feedback optimization targets static problems, i.e., steady-state optimization without dynamics on their own or sufficiently fast transients [26]. In addition, the methods previously applied in the context of particle accelerators are either unsuitable for continuous operation [6] or require extensive training on simulations [7].

The specific scheme of feedback optimization that the current paper builds upon has been proposed in Ref. [16]. Instead of projecting intermediate steps onto the feasible set or adding a penalty for constraint violation, the authors propose to solve a constrained linearization of the original problem. The attractiveness of their scheme stems from its guaranteed convergence even for non-convex problems and few oscillations in the transients even if operating close to the constraints. However, while [16] includes a metric in its quadratic program, the implications of choosing different metrics are only raised in passing.

After the introduction, Section 2 states the general optimization problem considered in this paper, reviews the concept of feedback optimization, and proposes a particular approach to include second-order information in the optimization problem. This is followed by a discussion of a scheme for data-driven sensitivity estimation for feedback optimizers in Section 3. The main content of the paper is presented in Section 4, which introduces the beamline position control problem at the European X-ray free-electron laser (EuXFEL) particle accelerator as an example, and evaluates feedback optimization with and without known sensitivities as a solution. Finally, Section 5 concludes the paper.

1.1 Contributions

The contributions of this paper are twofold:

1. Section 2.3 proposes an approximation scheme for the Hessian matrix of the optimization problem that can be evaluated from first-order derivative information of the underlying plant. Using this approximation, it becomes possible to take advantage of the Hessian for optimization in model-free settings, significantly improving the convergence speed of the algorithm.
2. We introduce the beam position control problem in particle accelerators as an application of feedback optimization in Section 4 that is challenging to solve with other methods such as Bayesian optimization.

1.2 Notation

We use I_n to denote the $n \times n$ identity matrix. In addition, $\mathbb{S}_+^n$ is the set of symmetric $n \times n$ matrices that are positive definite. For vectors $z \in \mathbb{R}^n$, $\|z\|_M := \sqrt{z^\top M z}$ denotes the weighted 2-norm with weight $M \in \mathbb{S}_+^n$. If the Euclidean norm with $M = I_n$ is used, the subscript is dropped for conciseness. Finally, $M_1 \otimes M_2$ is the Kronecker product of the matrices M_1 and M_2, and vec denotes the column-wise vectorization operator, i.e., vec(M) is the vector consisting of the stacked columns of the matrix M.

2 Feedback control for steady-state optimization

2.1 Problem statement

Consider a stable physical plant with input $u \in \mathbb{R}^p$ and output $y \in \mathbb{R}^n$, and a twice continuously differentiable cost function $\Phi\colon \mathbb{R}^p \times \mathbb{R}^n \to \mathbb{R}$ mapping u and y into a scalar cost. Our goal is to drive the plant to a steady-state pair (u^*, y^*) that minimizes Φ.

More precisely, we consider the physical plant to be expressed by an unknown or at least uncertain input-output steady-state map $h\colon \mathbb{R}^p \to \mathbb{R}^n$ such that $y = h(u)$ holds for all steady-state pairs (u, y). Furthermore, suppose there are polyhedral constraints $Au \leq b$ and $Cy \leq d$ on the admissible inputs and outputs, described by appropriately sized matrices A, C and vectors b, d, respectively, where we assume that the constraints are decoupled for simplicity. The desired steady-state pair (u^*, y^*) is then obtained from the solution of the optimization problem

$$\min_{u,y} \quad \Phi(u, y) \tag{1a}$$

$$\text{s.t.} \quad y = h(u), \tag{1b}$$

$$Au \leq b, \tag{1c}$$

$$Cy \leq d. \tag{1d}$$

Note that the cost Φ is a design parameter meaning that we assume perfect knowledge of the function.

There exist a variety of approaches for obtaining *local* solutions to optimization problems in the form of (1), e.g., projected gradient descent or sequential quadratic programming [1], while finding *global* solutions is generally intractable due to non-convexity. These methods can be roughly categorized to belong to one of two groups: On the one hand, there are optimization methods that perform a feedforward offline optimization based on an *a priori* known model of the plant. However, since no information from the physical plant is taken into account, the obtained input u will typically be suboptimal when actually applied. On the other hand, online methods like the Nelder-Mead algorithm or Bayesian optimization [1] use output measurements to find the optimal input but can only partially include available knowledge of the plant.

2.2 Feedback optimization

In its current formulation, the equality constraint (1b) implies that the output y can be eliminated from the optimization problem by substitution, leading to the new cost function $\tilde{\Phi}(u) := \Phi(u, h(u))$ in terms of only the input u. As such, measurements from the physical plant are not taken into account.

Instead, feedback can be included in the optimization process by utilizing measurements of the output y in place of knowledge of the steady-state map h [9]. The idea is to take advantage of the chain rule to calculate the Jacobian of the composite function $\tilde{\Phi}$ as

$$\frac{\partial \tilde{\Phi}}{\partial u}(u) = \frac{\partial \Phi}{\partial u}\big(u, h(u)\big) + \frac{\partial \Phi}{\partial y}\big(u, h(u)\big)\frac{\partial h}{\partial u}(u), \tag{2}$$

followed by a substitution of $h(u)$ by y. Note, that for computing the derivative of the composite cost $\tilde{\Phi}$ it is therefore only necessary to know the Jacobian $\frac{\partial h}{\partial u}$, not the full steady-state map h.

Based on this reformulation [16], implements the control law $u_{k+1} = u_k + \rho\kappa(u_k, y_k)$ with step size $\rho > 0$ and control update $\kappa\colon \mathbb{R}^p \times \mathbb{R}^n \to \mathbb{R}^p$, where u_k is the plant input after k steps and y_k is its corresponding measured output. For initialization, a suitable (with respect to the constraints) input u_0 can be chosen arbitrarily. At every step, the control update κ is determined by solving the parametric quadratic program

$$\kappa(u, y) := \arg\min_{v \in \mathbb{R}^p} \quad \left\| v + G^{-1}(u, y) J^\top(u, y) \right\|^2_{G(u,y)} \tag{3a}$$

$$\text{s.t.} \quad A(u + \rho v) \leq b, \tag{3b}$$

$$C\left(y + \rho \frac{\partial h}{\partial u}(u) v\right) \leq d, \tag{3c}$$

where $v \in \mathbb{R}^p$ is the optimization variable that can be thought of as the next step direction, $G\colon \mathbb{R}^p \times \mathbb{R}^n \to \mathbb{S}^p_+$ is a parameter-dependent weight, and

$$J(u, y) := \frac{\partial \Phi}{\partial u}(u, y) + \frac{\partial \Phi}{\partial y}(u, y)\frac{\partial h}{\partial u}(u)$$

is the Jacobian (2) of the composite cost evaluated at the measured output y instead of the steady-state map $h(u)$. As such, the only model information required to evaluate (3) is the Jacobian $\frac{\partial h}{\partial u}$. An approach without any knowledge of the plant is discussed in Section 3.

The structure of problem (3) is similar to other algorithms for solving nonlinear optimization problems but has the special property that it is partially evaluated at the measured output y. In particular, the weight G takes the role of the approximated Hessian that is common in quasi-Newton schemes [1]. For example, with $G(u, y) \equiv I$ and strictly inside the feasible set, i.e., both constraints inactive, the optimization variable v corresponds to the negative gradient.

Under suitable assumptions, i.e., feasibility of the problem, regularity of the constraint set, and a sufficiently small step size ρ, [16, Theorem 3] shows that the incremental control law drives the plant into a local minimum of the composite cost function $\tilde{\Phi}$. However, the result does not discuss the convergence speed of the method, especially with respect to the weight G. For iterative optimization algorithms, it is well known that including second-order information in form of the Hessian (or an approximation thereof) can drastically increase the speed of convergence close to the optimum, e.g., quasi-Newton methods or the Levenberg-Marquardt algorithm [1]. Hence, while convergence can be guaranteed also for $G(u, y) = I_p$, it is highly desirable to include second-order information when solving (3) if available.

2.3 Evaluating and approximating the Hessian

In the previous section, it was discussed how the chain rule can be employed to partially evaluate the Jacobian of the composite cost $\tilde{\Phi}$ at a measured output y. Importantly, the terms in (2) can be evaluated without knowledge of the full steady-state map h but only its Jacobian. However, no analogous steps have been employed on the Hessian, and the original formulation of the optimization problem with the weight G only depending on the input u does not allow for its evaluation in terms of the output y.

As an augmentation of optimization scheme discussed in Section 2.2, we therefore propose to perform a similar partial evaluation at the measured output y for the Hessian. Taking the Jacobian of $\left(\frac{\partial\tilde{\Phi}}{\partial u}\right)^{\top}$, we obtain

$$\begin{aligned}\frac{\partial^2\tilde{\Phi}}{\partial u^2}(u) = {} & \left(\frac{\partial h}{\partial u}(u)\right)^{\top}\frac{\partial^2\Phi}{\partial y^2}\big(u, h(u)\big)\frac{\partial h}{\partial u}(u)\\ & + \frac{\partial^2\Phi}{\partial u^2}\big(u, h(u)\big) + \frac{\partial^2\Phi}{\partial u \partial y}\big(u, h(u)\big)\frac{\partial h}{\partial u}(u)\\ & + \left(\frac{\partial h}{\partial u}(u)\right)^{\top}\frac{\partial^2\Phi}{\partial y \partial u}\big(u, h(u)\big)\\ & + \frac{\partial\Phi}{\partial y}\big(u, h(u)\big)\frac{\partial^2 h}{\partial u^2}(u). \end{aligned} \tag{4}$$

Note that $\frac{\partial^2 h}{\partial u^2}$ in the last term is a rank 3 tensor with dimension $n \times p \times p$ since h is a $\mathbb{R}^n$-valued function. Since we can again assume perfect knowledge of the cost function Φ, it is sufficient to know the Jacobian $\frac{\partial h}{\partial u}$ in order to evaluate the first four terms of (4), the knowledge of which is already assumed to solve problem (3). Furthermore, the reformulation allows for substituting the measured output y for the evaluation of the steady-state map $h(u)$. Hence, only the final term of (4) requires information beyond the Jacobian of h to compute, which is in particular its Hessian $\frac{\partial^2 h}{\partial u^2}$.

3 Adaptive feedback using sensitivity estimation

Section 2 shows how feedback can be included in an iterative algorithm to optimize the steady-state of a physical plant, and how existing approaches can be augmented by second-order information. However, as discussed in Ref. [9], achieving good convergence speed relies on having accurate knowledge of the Jacobian $\frac{\partial h}{\partial u}$. Especially for physical plants that are difficult to model using traditional methods, it would therefore be attractive to learn a model from input-output measurements.

One approach to obtain such a learned model is via recursive least squares estimation as proposed in Refs. [23], [24]. The idea is to have a recursive estimator, e.g., a Kalman filter, running with the plant in an online fashion. From the incremental changes applied to the plant, either by a controller running in parallel or on purpose for the estimation, it is then possible to extract the sensitivity, i.e., the Jacobian, of the steady-state map h at the current operating point of the plant. In this way, the estimator will learn the desired Jacobian with gradually increasing accuracy [9]. A particularly attractive feature of this approach is that *a priori* model information can be included in the filter as its initial state. Therefore, it is possible to refine an available approximate plant model online.

The essential idea behind the recursive estimator is to approximate the steady-state map through a time-varying linear system and utilize the rich literature on linear filtering. As the first step, consider its first-order Taylor expansion

$$\Delta y_k \approx \frac{\partial h}{\partial u}(u_k)\Delta u_k \tag{5}$$

of the steady-state map h with respect to the current input u_k, where the incremental input and output are defined as $\Delta u_k := u_{k+1} - u_k$ and $\Delta y_k := y_{k+1} - y_k$, respectively. Using the properties of the matrix vectorization operator and the Kronecker product, (5) can be rewritten as

$$\Delta y_k \approx \left(\Delta u_k^{\top} \otimes I_n\right)\mathrm{vec}\left(\frac{\partial h}{\partial u}(u_k)\right), \tag{6}$$

which contains the Jacobian in vectorized form. Defining $x_k := \mathrm{vec}\left(\frac{\partial h}{\partial u}(u_k)\right)$ for the Jacobian and $U_k := \Delta u_k^{\top} \otimes I_n$ for the incremental input, (6) can be approximated as the output of a linear time-varying system taking the form

$$x_{k+1} = x_k + v_k \tag{7a}$$

$$\Delta y_k = U_k x_k + w_k \tag{7b}$$

with disturbances v_k and w_k. The behavior in (7a) is chosen since the Jacobian is fixed for constant operating points and slow changes can be captured in the process disturbance v_k.

Based on the artificial linear system (7), we can design a Kalman filter for estimating x_k and thus the Jacobian. Under the hypothesis that v_k and w_k are independent zero-mean white Gaussian noise with covariance Q_e and R_e, the filter update equations are given by

$$\hat{x}_{k+1} = \hat{x}_k + K_k\left(\Delta y_k - U_k \hat{x}_k\right) \tag{8a}$$

$$\Sigma_{k+1} = \left(I - K_k U_k\right)\Sigma_k + Q_\mathrm{e} \tag{8b}$$

$$K_k = \Sigma_k U_k^\top \left(R_\mathrm{e} + U_k \Sigma_k U_k^\top\right)^{-1} \tag{8c}$$

with the estimate $\hat{x}_k$ and its covariance Σ_k. Under a persistency of excitation condition on Δu_k, [23, Proposition 1] proves that the filter asymptotically converges towards an unbiased estimate of the Jacobian with bounded variance. Note, however, that the hypothesis on v_k and w_k is restrictive and might not be satisfied in practice and that [23, Proposition 1] therefore only provides an indication for the convergence properties of the above filter.

Finally, since the feedback optimization problem (3) requires only knowledge of the Jacobian $\frac{\partial h}{\partial u}(u_k)$, sensitivity estimation can be utilized for model-free closed-loop optimization. That is, at step k, the current estimate $\hat{x}$ is reshaped into the $n \times p$ matrix $\nabla \tilde{h}_k$ which is employed for optimization instead of $\frac{\partial h}{\partial u}(u_k)$. As such, an approximation of problem (3) is solved, leading to the update direction $\tilde{\kappa}(u_k, y_k)$ being applied to the plant through the control law

$$u_{k+1} = u_k + \rho \tilde{\kappa}(u_k, y_k) + z_k,$$

where z_k is a random excitation signal that is added to ensure the persistency excitation condition is satisfied. With sufficiently small step size ρ, the closed loop of the estimator and optimizer converges to a mean-square bounded region around the optimal input u^* [23, Proposition 1].

Since the estimator for the Jacobian is based on a first-order Taylor expansion, one could hypothesize that an analogous estimator can be formulated for the Hessian using more series terms. As such, we compute the second-order Taylor expansion for the i-th component of the steady-state map h as

$$\Delta y_k^i \approx \frac{\partial h_i}{\partial u}(u_k)\Delta u_k + \frac{1}{2}\Delta u_k^\top \frac{\partial^2 h_i}{\partial u^2}(u_k)\Delta u_k. \tag{9}$$

Importantly, note that (9) is affine in both the Jacobian $\frac{\partial h_i}{\partial u}$ *and* the Hessian $\frac{\partial^2 h_i}{\partial u^2}$. Therefore, we obtain an extended artificial linear system in the form of (7) with state

$$\tilde{x}_k := \begin{bmatrix} \operatorname{vec}\left(\dfrac{\partial h}{\partial u}(u_k)\right) \\ \operatorname{vec}\left(\dfrac{\partial^2 h}{\partial u^2}(u_k)\right) \end{bmatrix}$$

and output matrix

$$\tilde{U}_k := \begin{bmatrix} 1 & \Delta u_k^\top \end{bmatrix} \otimes \Delta u_k^\top \otimes I_n.$$

Unfortunately, vectors in the form $\left(\begin{bmatrix} 1 & v^\top \end{bmatrix} \otimes v^\top\right)^\top$ with $v \in \mathbb{R}^p$ only span a p-dimensional subspace. As such, no excitation Δu_k is rich enough to provide enough information for estimating the extended state $\tilde{x}_k$, and the sensitivity estimator (8) cannot be employed to approximate the Hessian $\frac{\partial^2 h_i}{\partial u^2}$.

4 Feedback optimization for control of particle accelerators

Particle Accelerators are complex machines that possess many dynamically coupled and interacting control loops. One feature is their large timescale separation: While the particles are moving at close to the speed of light and are thus passing through the machine in the range of microseconds, adjusting the steering magnets may take in the order of seconds. As such, they are a prime target for application of feedback optimization.

There are multiple kinds of particle accelerators, most importantly circular and linear accelerators, with different characteristics. In this paper, we will focus in particular on the linear accelerator of EuXFEL. Operated by DESY in Hamburg, Germany, the EuXFEL accelerates electrons to energies up to 17.5 GeV in order to produce X-ray pulses for multiple experimental stations [27]. This is achieved in pulsed-mode operation, accelerating up to 2,700 electron bunches per pulse at a repetition rate of 4.5 MHz with a pulse rate of 10 Hz for a total of 27,000 bunches per second. A schematic overview of the facility is given in Figure 1.

For this paper, we are interested especially in one section of the EuXFEL, the main electron beam dump line shown in the bottom right of Figure 1. Located at the end of the main accelerating section, the purpose of the beam dump is to absorb the energy of electrons that are not intended for photon emission. The main beam dumps are designed to continuously and safely stop an electron beam with up to 300 kW, but it is crucial to distribute the load on

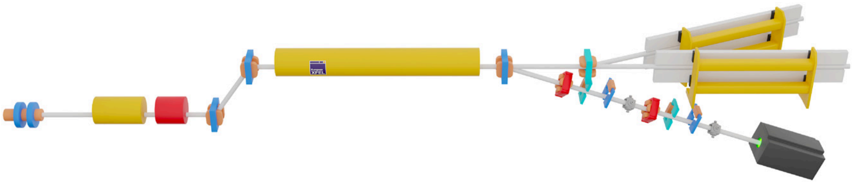

Figure 1: A schematic overview of the EuXFEL facility. On the left is the electron source, the yellow and red tubes symbolize the superconducting accelerating modules. After the main 1.5 km long accelerating section, the beamline branches out to the photon-emitting undulators at the top and the main beam dump at the bottom. Along the beamline, dipole magnets (blue and cyan) are used for steering, while quadrupole magnets (red) focus the beam.

the dump surface, which is accomplished using a pair of sweeping magnets. However, these magnets rely on a centered and leveled incoming beam, otherwise their deflection can misdirect the electrons off the intended surface and overheat the dump.

Steering the beam is accomplished using a sequence of dipole and quadrupole magnets situated along the beamline as indicated in Figure 1 in the section leading to the dump. With the dipoles, it is possible to steer the beam in the horizontal (shown in cyan) and vertical plane (blue), and the quadrupoles (red) can focus and defocus the beam. Additionally, quadrupoles also steer the beam if it enters the magnet off-center. In between the magnets, beam position monitors (BPMs) are placed to measure the transverse position of the beam. A brief description of the beam dynamics is provided in the next section.

4.1 Linear beamline tracking

A particle in the accelerator can be described relative to a reference particle by a 6 + 1-dimensional state

$$\xi = \begin{bmatrix} x & p_x & y & p_y & \tau & \delta & 1 \end{bmatrix}^\top,$$

where x and y are the transverse position in horizontal and vertical direction, p_x and p_y are the respective *normalized* momenta relative to the reference momentum p_0, and

$$\tau = c\Delta t \quad \delta = \frac{E - E_0}{c p_0}$$

are the longitudinal position and energy offset, respectively, with c being the speed of light. Finally, a constant 1 is added as the last component to cover affine effects in the dynamics.

In the context of this paper, three types of accelerator lattice elements are of special interest. We will therefore introduce a linear beamline model in the following, neglecting collective effects, e.g., space charge effects, and fringe fields at the element boundaries. Note, however, that the linear effects dominate at ultra-relativistic particle energies, such that the linear model is reasonable to apply to the particular EuXFEL section of interest, where the beam has already reached GeV-level energies [28]. A fast implementation of particle tracking routines including non-linear and inter-particle effects is provided by CHEETAH [29], [30], which will be employed for the simulations below.

4.1.1 Drift space

The simplest kind of element are *drift spaces*, straight sections of the beamline where particles travel without disturbance by electromagnetic fields. In reality, the particle bunches would be subject to internal fields, but these effects are neglected here. Drift spaces are characterized by their length l in m and their transfer map is

$$T_\mathrm{d} = \begin{bmatrix} 1 & l & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & l & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & -\frac{l}{\beta^2}\frac{1}{\gamma^2} & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}, \tag{10}$$

where $\gamma := \frac{E}{m_\mathrm{e}c^2}$ is the relativistic Lorentz factor with electron mass m_e and relativistic velocity factor $\beta := \sqrt{1 - \frac{1}{\gamma^2}}$. The formulation of the 7-dimensional transfer matrices is taken from Ref. [30] which is based on the results of Ref. [31].

4.1.2 Steering dipole magnet

The second kind of elements are *steering magnets* that deflect the particle beam at an adjustable angle α in either the horizontal or vertical plane. For a horizontal steering magnet, the transfer matrix is given by

$$T_{\mathrm{sh}} = \begin{bmatrix} 1 & l & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & \alpha \\ 0 & 0 & 1 & l & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & -\frac{l}{\beta^2}\frac{1}{\gamma^2} & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}, \qquad (11)$$

while vertical adjustments are described by the matrix

$$T_{\mathrm{sv}} = \begin{bmatrix} 1 & l & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & l & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & \alpha \\ 0 & 0 & 0 & 0 & 1 & -\frac{l}{\beta^2}\frac{1}{\gamma^2} & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}. \qquad (12)$$

The physical length of the magnet in longitudinal direction is again given by l.

4.1.3 Quadrupole

Finally, the third kind of elements are quadrupole magnets. In contrast to dipole magnets, quadrupoles do not deflect all particles at the same angle but act like lenses, focusing the beam in one (horizontal or vertical) plane and defocusing it in the other. They are parametrized by their length l and the focusing strength k in m^{-2}, which is positive for magnets that focus in the horizontal plane and negative for vertical focusing. Their transfer map is described by

$$T_{\mathrm{q}} = \begin{bmatrix} c_x & s_x & 0 & 0 & 0 & \frac{1-c_x}{\beta} & 0 \\ -ks_x & c_x & 0 & 0 & 0 & \frac{s_x}{\beta} & 0 \\ 0 & 0 & c_y & s_y & 0 & 0 & 0 \\ 0 & 0 & ks_y & c_y & 0 & 0 & 0 \\ \frac{s_x}{\beta} & \frac{1-c_x}{\beta} & 0 & 0 & 1 & r_{56} & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \qquad (13)$$

with

$$r_{56} = \frac{l - s_x}{\beta^2 k} - \frac{l}{\beta^2 \gamma^2}$$

and

$$s_{\mathrm{x}} = \mathrm{Re}\left(\frac{\sin(k_x l)}{k_x}\right) \qquad s_y = \mathrm{Re}\left(\frac{\sin(k_y l)}{k_y}\right)$$
$$c_x = \mathrm{Re}\big(\cos(k_x l)\big) \qquad c_y = \mathrm{Re}\big(\cos(k_y l)\big)$$
$$k_x = \sqrt{k} \qquad k_y = \sqrt{-k}.$$

In addition to this ideal transfer map, quadrupole magnets can be misaligned, i.e., their center point does not align with the reference particle, and tilted around the beam in the transverse plane, leading to undesired steering of the particle distribution. Such adjustments result in shifts and rotations of the frame of reference, which can be captured in additional matrices that have to be pre- and postmultiplied to T_{q}.

4.2 Dump beamline feedback with known sensitivity

Having introduced the relevant beamline elements and their purpose, we arrive at the specifics of the control problem at hand. Along the beamline section of the main electron dump, there are twelve steering magnets, six in the horizontal plane and six in the vertical plane, in alternating fashion. In between the magnets, ten BPMs are placed to determine the transverse position of the beam, each measuring the horizontal and vertical component. Furthermore, quadrupole magnets are placed along the beamline. Our goal is to find a tuple of steering angles $(\alpha_0, \ldots, \alpha_{12})$ that minimize the deviation of the particles from the center position of the BPMs.

The prevalent disturbance in the problem is the state of the particle beam entering the dump beamline section. Since the main purpose of the beam is generating photons, the particle positions at the entrance to the dump section are not controlled, causing the mean position to potentially drift over time. Furthermore, the alignment of the quadrupoles and BPMs with respect to the reference beam is difficult, presenting an uncertainty in the beamline model. The task of the optimizer is therefore to reject the disturbance and steer the beam to the center position of the BPMs. As such, we select the quadratic cost function

$$\Phi(u, y) = \|y\|^2 + \eta\|u\|^2$$

with tuning parameter $\eta > 0$, and set $\eta = 10^{-3}$ in the following. Note that, depending on the incoming beam, it may be impossible to center the beam at all BPMs due to the arrangement of the position monitors and magnets. In addition, the optimization problem is constrained to keep the beam within ±2 cm of the reference at the BPMs to prevent hitting the surrounding beam pipe.

For the first demonstration, we study the effect of including the Hessian in the optimization problem (3) as discussed in Section 2.3. However, since the curvature term $\frac{\partial^2 h}{\partial u^2}$ can be difficult to obtain, we propose to neglect the final component of (4) while calculating the Hessian and therefore set

$$G(u,y) = \frac{\partial^2\Phi}{\partial u^2}(u,y) + \left(\frac{\partial h}{\partial u}(u)\right)^{\top} \frac{\partial^2\Phi}{\partial y^2}(u,y)\frac{\partial h}{\partial u}(u) + \frac{\partial^2\Phi}{\partial u \partial y}(u,y)\frac{\partial h}{\partial u}(u) + \left(\frac{\partial h}{\partial u}(u)\right)^{\top} \frac{\partial^2\Phi}{\partial y \partial u}(u,y) \tag{14}$$

for the parameter-varying weight in the optimization problem. Note that $\frac{\partial^2 h}{\partial u^2} = 0$ in case of the dump beamline position feedback problem since the dipole response matrices T_{sh} and T_{sv} are linear in the steering angle α, such that no error is incurred by this approximation for the current scenario. Furthermore, $G(u,y)$ is guaranteed to be positive definite because the chosen Φ does not contain cross terms between u and y and satisfies $\eta > 0$.

In order to evaluate the difference in convergence speed between the two setups, we run 30 scenarios with randomly positioned incoming beams and quadrupole misalignment using the beam dynamics simulation in Cheetah and the quadratic program solver qpth [32]. Furthermore, we take advantage of the differentiability of Cheetah to get access to the sensitivity $\frac{\partial h}{\partial u}$ with the reference beam and perfect quadrupole alignment, *not* the actual incoming beam and misalignment. Both the beam center and the misalignment are sampled from zero mean normal distributions with a standard deviation of 1 mm for the beam and 250 μm for the magnets. All 30 scenarios are run twice, once with only first-order information, i.e., $G(u,y) = I$, and step size $\rho_{\mathrm{jac}} = 10^{-5}$ and once with the (approximated) Hessian as given in (14) and $\rho_{\mathrm{hes}} = 3 \times 10^{-1}$. In both cases, the step size has been manually tuned for fast convergence. Increasing the step size in the first-order case to above 10^{-4} renders the closed-loop unstable, leading to divergence of the algorithm. The deviation is evaluated as the root mean square error (RMSE) taken over all scenarios and summed over the BPMs and is plotted on a logarithmic scale in Figure 2.

As seen in Figure 2, the RMSE converges to its final value within 20 steps when employing the second-order information of the Hessian. On the other hand, the convergence is much slower when only using the Jacobian, obtaining an RMSE that is over a magnitude worse than the optimal value of the first case after 500 iterations., demonstrating the importance of including second-order information when available. Furthermore, Table 1 shows the time required for computing the derivatives, solving the quadratic program, and in total, with and without employing the Hessian matrix. Even though the variant including the Hessian matrix takes approximately twice as long to evaluate, it is well within the 100 ms sampling interval of the EuXFEL. As such, the improved convergence outweighs the increased computational demand.

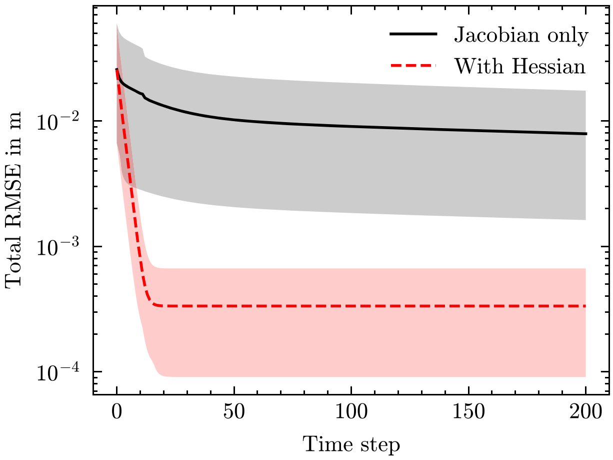

Figure 2: Comparison of the convergence speed of the closed-loop feedback optimization with purely first-order information (Jacobian only) and with the approximated Hessian in (14). The RMSE is taken over all BPMs and with 30 random perturbations of the incoming electron beam and quadrupole misalignments, while the shading indicates the best- and worst-case scenarios.

4.3 Online identification and control

For the second scenario, we assume that the sensitivity $\frac{\partial h}{\partial u}$ is unknown and has to be estimated as described in Section 3. The excitation signal z_k is taken as zero mean white Gaussian noise with standard deviation 10^{-7}. Furthermore, we choose G as given in (14) with the estimated sensitivity $\nabla\tilde{h}_k$ substituted for the Jacobian $\frac{\partial h}{\partial u}$ and use the step size $\rho = 3 \times 10^{-1}$. For the Kalman filter, the noise covariance

Table 1: Computation times of the feedback optimizer with and without Hessian matrix.

Hessian	Derivatives [ms]		Solve [ms]		Total [ms]	
	Median	Max	Median	Max	Median	Max
Without	0.17	0.22	0.69	2.42	0.87	2.61
With	7.39	7.95	1.08	1.78	8.50	9.19

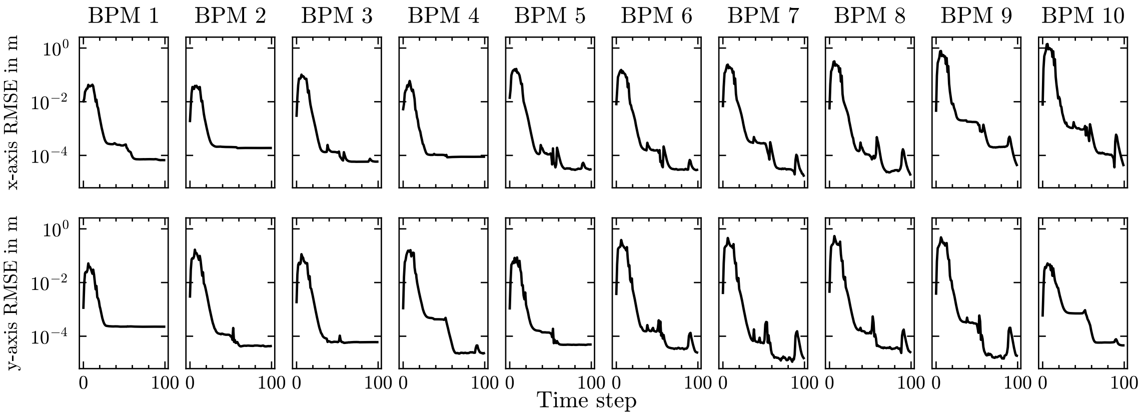

Figure 3: RMSE over 30 random incoming electron beams and quadrupole misalignment in the *x* and *y* component of all ten BPMs.

matrices are set as $Q_\mathrm{e} = 10I$ and $R_\mathrm{e} = 10^{-7}I$, respectively, where the magnitude of R_e has to be seen relative to the excitation z_k [9].

As before, we simulate 30 runs with randomly sampled incoming beam and quadrupole misalignment, taking the same samples as in the previous section for better comparability. In addition, the initial estimate $\hat{x}_0$ of the sensitivity is chosen at random for each run. The behavior of the RMSE for each position measurement is shown in Figure 3 on a logarithmic scale. Contrasting the previous scenario with known sensitivity $\frac{\partial h}{\partial u}$, the optimizer is initially steering the beam away from the BPM center positions due to the randomly initialized sensitivities. However, the RMSE starts improving after less than 20 iterations, quickly recovering the lost performance and converging to an improved solution within 50 steps.

This behavior demonstrates how using the Hessian (14) can improve the convergence speed even with estimated sensitivity, resulting in a better performance than the first-order case shown in the previous section and a similar final RMSE to the scenario with Hessian. This is visualized in Figure 4. For all 30 scenarios, the difference between the two schemes is not more than 20 µm in RMSE, and in every second scenario the difference is close to 0, indicating that the estimated sensitivity achieves an accuracy comparable to the case with known model.

For comparison, the same 30 scenarios have also been optimized using the Nelder-Mead algorithm which is routinely deployed at the EuXFEL through OCELOT [6]. The implementation is taken from the PYTHON SCIPY library [33], [34] and the resulting RMSE is shown in Figure 5. After the initial increase, both algorithms start optimizing at approximately the same RMSE. However, while the Nelder-Mead simplex algorithm is monotonically improving its solution, the feedback optimizer is converging at a much faster rate.

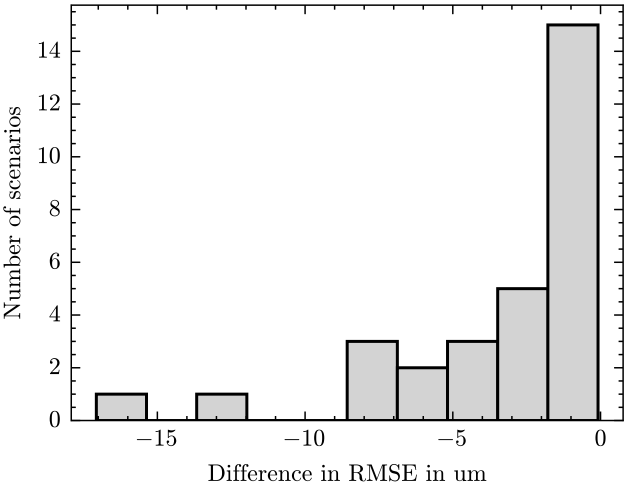

Figure 4: Histogram of the difference in final RMSE between optimization with known model from Section 4.2 and with the sensitivity estimation from Section 4.3. Negative values indicate that the estimated sensitivity achieved worse accuracy.

Finally, consider Figure 6, which shows the Frobenius norm of the difference between the actual Jacobian $\frac{\partial h}{\partial u}$ of the accelerator and the estimated sensitivity $\nabla\tilde{h}_k$. Similar to the RMSE of the position, the sensitivity error shows a fast decrease at the beginning. This is caused by the strong excitation of the plant due to the feedback optimizer searching for the optimal input. The filter continues to improve its estimate based on the excitation z_k once the initial plateau is reached by the optimizer, albeit at a slower pace, before converging after 28,000 iterations. Note, however, that the final error and the convergence speed depend on the excitation z_k and the chosen covariance matrices Q_e and R_e.

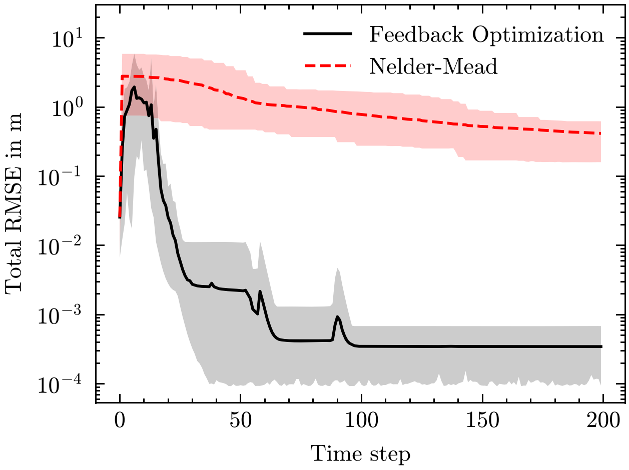

Figure 5: Comparison of the total RMSE between feedback optimization and the Nelder-Mead algorithm. The shaded areas indicate the best and worst case of the 30 scenarios.

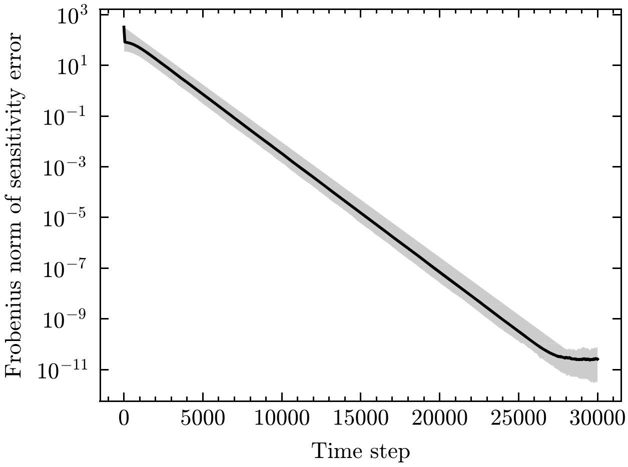

Figure 6: Frobenius norm of the difference between the estimated and actual sensitivity plotted over time. The solid line indicates the RMSE, while the shading visualizes the best- and worst-case.

5 Conclusions

In this paper, we have introduced the beam position control problem in particle accelerators as an application of feedback optimization. Furthermore, a particular approximation of the Hessian of the resulting quadratic optimization problem is proposed. The two presented scenarios – model-based with known steady-state sensitivity and model-free with sensitivity estimation based on the Kalman filter – demonstrate that feedback optimization is a highly variable method for controlling particle accelerators. Even with no initial information about the accelerator, the proposed approach is able to improve the beam positioning within 20 steps and converges to a final with comparable accuracy to the model-based case in less than 100 iterations. Moreover, since the convergence is faster than in the scenario without second-order information considered in Section 4.2, the simulations show that the Hessian approximation (14) is an essential component in achieving the desired convergence speed in the model-free setting.

Research ethics: Not applicable.
Informed consent: Not applicable.
Author contributions: All authors have accepted responsibility for the entire content of this manuscript and approved its submission.
Use of Large Language Models, AI and Machine Learning Tools: None declared.
Conflict of interest: The authors state no conflict of interest.
Research funding: We gratefully acknowledge funding by the EuXFEL R&D project “RP-513: Learning Based Methods”.
Data availability: Not applicable.

References

[1] J. Nocedal and S. J. Wright, *Numerical Optimization*, 2nd ed. New York, NY, Springer, 2006.

[2] K. Zhou, J. C. Doyle, and K. Glover, *Robust and Optimal Control*, Upper Saddle River, NJ, Prentice Hall, 1996, p. 596.

[3] J. B. Rawlings, D. Q. Mayne, and M. Diehl, *Model Predictive Control: Theory, Computation, and Design*, 2nd ed. Santa Barbara, California, Nob Hill Publishing, 2017.

[4] B. Shahriari, K. Swersky, Z. Wang, R. P. Adams, and N. de Freitas, “Taking the human out of the loop: a review of Bayesian optimization,” *Proc. IEEE*, vol. 104, no. 1, pp. 148–175, 2016.

[5] R. Roussel, *et al*., “Bayesian optimization algorithms for accelerator physics,” *Phys. Rev. Accel. Beams*, vol. 27, no. 8, 2024, Art. no. 084801. https://doi.org/10.1103/physrevaccelbeams.27.084801.

[6] S. Tomin, I. Agapov, W. Decking, G. Geloni, M. Scholz, and I. Zagorodnov, “On-line optimization of european XFEL with OCELOT,” in *16th Int. Conf. on Accelerator and Large Experimental Control Systems*, Geneva, Switzerland, JACOW, 2018, pp. 1038–1042.

[7] J. Kaiser, *et al*., “Reinforcement learning-trained optimisers and Bayesian optimisation for online particle accelerator tuning,” *Sci. Rep.*, vol. 14, no. 1, 2024, Art. no. 15733. https://doi.org/10.1038/s41598-024-66263-y.

[8] A. Simonetto, E. Dall’Anese, S. Paternain, G. Leus, and G. B. Giannakis, “Time-varying convex optimization: time-structured algorithms and applications,” *Proc. IEEE*, vol. 108, no. 11, pp. 2032–2048, 2020.

[9] A. Hauswirth, Z. He, S. Bolognani, G. Hug, and F. Dörfler, “Optimization algorithms as robust feedback controllers,” *Ann. Rev. Control*, vol. 57, 2024, Art. no. 100941. https://doi.org/10.1016/j.arcontrol.2024.100941.

[10] J. C. Doyle, B. A. Francis, and A. R. Tannenbaum, *Feedback Control Theory*, New York, NY, Macmillan Publishing Co., 1990.

[11] F. Blanchini, G. Fenu, G. Giordano, and F. A. Pellegrino, “Model-free plant tuning,” *IEEE Trans. Automat. Control*, vol. 62, no. 6, pp. 2623–2634, 2017.

[12] M. Colombino, J. W. Simpson-Porco, and A. Bernstein, “Towards robustness guarantees for feedback-based optimization,” in *2019 IEEE 58th Conference on Decision and Control (CDC)*, 2019, pp. 6207–6214.

[13] A. Bernstein, E. Dall’Anese, and A. Simonetto, “Online primal-dual methods with measurement feedback for time-varying convex optimization,” *IEEE Trans. Signal Process.*, vol. 67, no. 8, pp. 1978–1991, 2019.

[14] A. Jokic, M. Lazar, and P. P. J. van den Bosch, “On constrained steady-state regulation: dynamic KKT controllers,” *IEEE Trans. Automat. Control*, vol. 54, no. 9, pp. 2250–2254, 2009.

[15] M. Colombino, E. Dall’Anese, and A. Bernstein, “Online optimization as a feedback controller: stability and tracking,” *IEEE Trans. Control Netw. Syst.*, vol. 7, no. 1, pp. 422–432, 2020.

[16] V. Häberle, A. Hauswirth, L. Ortmann, S. Bolognani, and F. Dörfler, “Non-convex feedback optimization with input and output constraints,” *IEEE Control Syst. Lett.*, vol. 5, no. 1, pp. 343–348, 2021.

[17] C. E. Garcia and M. Morari, “Optimal operation of integrated processing systems: part II: closed-loop on-line optimizing control,” *AIChE J.*, vol. 30, no. 2, pp. 226–234, 1984.

[18] S. Costello, G. François, D. Bonvin, and A. Marchetti, “Modifier adaptation for constrained closed-loop systems,” *IFAC Proc. Vol.*, vol. 47, no. 3, pp. 11080–11086, 2014.

[19] F. P. Kelly, A. K. Maulloo, and D. K. H. Tan, “Rate control for communication networks: sshadow prices, proportional fairness and stability,” *J. Oper. Res. Soc.*, vol. 49, no. 3, pp. 237–252, 1998.

[20] N. Li, C. Zhao, and L. Chen, “Connecting automatic generation control and economic dispatch from an optimization view,” *IEEE Trans. Control Netw. Syst.*, vol. 3, no. 3, pp. 254–264, 2016.

[21] D. K. Molzahn, *et al*., “A survey of distributed optimization and control algorithms for electric power systems,” *IEEE Trans. Smart Grid*, vol. 8, no. 6, pp. 2941–2962, 2017.

[22] G. Bianchin, M. Vaquero, J. Cortés, and E. Dall’Anese, “Online stochastic optimization for unknown linear systems: data-driven controller synthesis and analysis,” *IEEE Trans. Automat. Control*, vol. 69, no. 7, pp. 4411–4426, 2024.

[23] M. Picallo, L. Ortmann, S. Bolognani, and F. Dörfler, “Adaptive real-time grid operation via online feedback optimization with sensitivity estimation,” *Elec. Power Syst. Res.*, vol. 212, 2022, Art. no. 108405, https://doi.org/10.1016/j.epsr.2022.108405.

[24] A. D. Domínguez-García, M. Zholbaryssov, T. Amuda, and O. Ajala, “An online feedback optimization approach to voltage regulation in inverter-based power distribution networks,” in *American Control Conference*, 2023, pp. 1868–1873.

[25] M. Diehl, H. G. Bock, and J. P. Schlöder, “A real-time iteration scheme for nonlinear optimization in optimal feedback control,” *SIAM J. Control Optim.*, vol. 43, no. 5, pp. 1714–1736, 2005.

[26] A. Hauswirth, S. Bolognani, G. Hug, and F. Dörfler, “Timescale separation in autonomous optimization,” *IEEE Trans. Automat. Control*, vol. 66, no. 2, pp. 611–624, 2021.

[27] W. Decking, *et al*., “A MHz-repetition-rate hard X-ray free-electron laser driven by a superconducting linear accelerator,” *Nat. Photonics*, vol. 14, no. 6, pp. 391–397, 2020.

[28] H. Wiedemann, *Particle Accelerator Physics (Graduate Texts in Physics)*, 4th ed. Cham, Springer International Publishing, 2015.

[29] O. Stein, J. Kaiser, and A. Eichler, “Accelerating linear beam dynamics simulations for machine learning applications,” in *Proceedings of the 13th International Particle Accelerator Conference*, 2022.

[30] J. Kaiser, C. Xu, A. Eichler, and A. Santamaria Garcia, “Bridging the gap between machine learning and particle accelerator physics with high-speed, differentiable simulations,” *Phys. Rev. Accel. Beams*, vol. 27, no. 5, 2024, Art. no. 054601.

[31] K. L. Brown, “A first and second order matrix theory for the design of beam transport systems and charged particle spectrometers,” *Adv. Part. Phys.*, vol. 1, 1968.

[32] B. Amos and J. Z. Kolter, “OptNet: differentiable optimization as a layer in neural networks,” in *International Conference on Machine Learning, ser. ICML’17*, Sydney, NSW, Australia, 2017, pp. 136–145.

[33] P. Virtanen, *et al*., “SciPy 1.0: fundamental algorithms for scientific computing in python,” *Nat. Methods*, vol. 17, pp. 261–272, 2020.

[34] F. Gao and L. Han, “Implementing the Nelder-Mead simplex algorithm with adaptive parameters,” *Comput. Opt. Appl.*, vol. 51, no. 1, pp. 259–277, 2012.

Bionotes

Christian Hespe
Deutsches Elektronen-Synchrotron DESY,
Notkestraße 85, 22607 Hamburg, Germany
christian.hespe@desy.de

Christian Hespe received the B.Sc. and M.Sc. degrees in electrical engineering from the Hamburg University of Applied Sciences, Germany, and the Hamburg University of Technology, Germany, in 2018 and 2020. In 2020, he started pursuing a Ph.D. degree, before joining Deutsches Elektronen-Synchrotron DESY as a research scientist. His current research is focussed on learning-based and optimization-driven control.

Jan Kaiser
Deutsches Elektronen-Synchrotron DESY,
Notkestraße 85, 22607 Hamburg, Germany
jan.kaiser@desy.de

Jan Kaiser received his M.Sc. degree in Computer Science from Hamburg University of Technology in 2020. In 2020 he started his Ph.D. as a Doctoral Researcher and later Research Scientist at Deutsches Elektronen-Synchrotron DESY. His research focusses on reinforcement learning and differentiable high-speed simulations towards autonomous particle accelerator operations.

Jannis Lübsen
Institute of Control Systems, Hamburg University of Technology, Harburger Schloßstraße 22a, 21079 Hamburg, Germany
jannis.luebsen@tuhh.de

Jannis Lübsen received the B.Sc. and M.Sc. degree in mechatronics from the University of Technology in Hamburg in 2020 and 2022, respectively. He joined the Institute of Control Systems at the Hamburg University of Technology as research assistant in 2022, and is currently engaged in pursuing his doctoral degree. His research interests include data-driven optimization, and applied operator theory with application in particle accelerators.

Frank Mayet
Deutsches Elektronen-Synchrotron DESY, Notkestraße 85, 22607 Hamburg, Germany
frank.mayet@desy.de

Frank Mayet received his M.Sc. degree in physics in 2012 and his Ph.D. degree in physics in 2019, both from the University of Hamburg in collaboration with DESY Hamburg, Germany. From 2013 to 2016, he worked as a software engineer outside of academia. Until 2024, he was a postdoctoral researcher at the ARES accelerator, a dedicated R&D facility at DESY Hamburg. Currently he is senior staff scientist in the European XFEL accelerator operations group at DESY Hamburg. His work focuses on the development and implementation of advanced accelerator control system and automation tools, with an emphasis on machine learning and artificial intelligence applications for optimizing facility operations.

Matthias Scholz
Deutsches Elektronen-Synchrotron DESY, Notkestraße 85, 22607 Hamburg, Germany
matthias.scholz@desy.de

Matthias Scholz received his Ph.D. in physics from the University of Hamburg in 2013. Since 2017, he has been a staff scientist at Deutsches Elektronen-Synchrotron (DESY), where he is responsible for the daily operation of the accelerator at the Free Electron Laser European XFEL. In addition to his operational role, he is a member of the DESY Scientific Committee and the European XFEL Operation Board.

Annika Eichler
Deutsches Elektronen-Synchrotron DESY, Notkestraße 85, 22607 Hamburg, Germany; and Institute of Control Systems, Hamburg University of Technology, Harburger Schloßstraße 22a, 21079 Hamburg, Germany
annika.eichler@tuhh.de

Annika Eichler has studied General Engineering Science at the Hamburg University of Technology and received her Diploma in Mechatronics with major in Control in 2010. She was joining the Institute of Control at the Hamburg University of Technology as research assistant and obtained her Ph.D. in 2015. From 2015 to 2019 she was with the Institute of Control at the ETH Zurich, Switzerland, starting as Postdoc and then as senior scientist. In 2019 she joined the Deutsches Elektronen Synchrotron DESY in Hamburg to build up a research group in intelligent process control. Since 2022 she has a full professorship at the Hamburg University of Technology in collaboration with DESY, associated to the Institute of Control. Her research interests include distributed, data-driven and robust control and optimization as well as fault diagnosis with application mainly in particle accelerators.