Article

Adaptive Machine Learning for Robust Diagnostics and Control of Time-Varying Particle Accelerator Components and Beams

Alexander Scheinker [1,†,*]

[1] Los Alamos National Laboratory, Los Alamos, NM 87545, USA
[*] Correspondence: ascheink@lanl.gov
[†] Current address: Los Alamos National Laboratory, Los Alamos, NM 87545, USA

Abstract: Machine learning (ML) is growing in popularity for various particle accelerator applications including anomaly detection such as faulty beam position monitor or RF fault identification, for non-invasive diagnostics, and for creating surrogate models. ML methods such as neural networks (NN) are useful because they can learn input-output relationships in large complex systems based on large data sets. Once they are trained, methods such as NNs give instant predictions of complex phenomenon, which makes their use as surrogate models especially appealing for speeding up large parameter space searches which otherwise require computationally expensive simulations. However, quickly time varying systems are challenging for ML-based approaches because the actual system dynamics quickly drifts away from the description provided by any fixed data set, degrading the predictive power of any ML method, and limits their applicability for real time feedback control of quickly time-varying accelerator components and beams. In contrast to ML methods, adaptive model-independent feedback algorithms are by design robust to un-modeled changes and disturbances in dynamic systems, but are usually local in nature and susceptible to local extrema. In this work, we propose that the combination of adaptive feedback and machine learning, adaptive machine learning (AML), is a way to combine the global feature learning power of ML methods such as deep neural networks with the robustness of model-independent control. We present an overview of several ML and adaptive control methods, their strengths and limitations, and an overview of AML approaches. A simple code for the adaptive control algorithm used here can be downloaded from: https://github.com/alexscheinker/ES_adaptive_optimization

Keywords: machine learning; adaptive control; time-varying systems

1. Introduction

Machine learning (ML) [1] tools such as neural networks (NN) [2], Gaussian processes (GP) [3], and reinforcement learning (RL) in which NNs are incorporated to represent system models and optimal feedbacks [4], have been growing in popularity for particle accelerator applications. Although these methods have been around for decades, their recent growth in popularity can be attributed to recent growth in computing power with high performance computers and especially graphics processing units (GPUs) becoming very inexpensive. Also, powerful and easy to use software packages such as tensorflow are now freely available for anyone to easily develop their own sophisticated ML tools specifically tailored to their accelerator problems.

Recent ML applications for accelerators include ML-enhanced genetic optimization [5], utilizing surrogate models for simulation-based optimization studies and for estimating beam characteristics [6–10], Bayesian and GP approaches for accelerator tuning [11–16], various applications at the LHC including optics corrections and detecting faulty beam position monitors [17–19], powerful polynomial chaos expansion-based surrogate models for uncertainty quantification have been developed [20], and RL tools have been developed for online accelerator optimization [21–25]. One challenge faced by many ML approaches is the fact that as accelerators and their beams change with time, the ML models that were trained with previously collected data are no longer accurate because they are being applied to a different system than the one which they have been trained for. In order to

doi:10.20944/preprints202103.0301.v1

provide accurate control and diagnostics in the presence of time varying systems a real time feedback adaptive ML approach is required.

Recently, novel adaptive feedback algorithms have been developed which are able to tune large groups of parameters simultaneously based only on noisy scalar measurements with analytic proofs of convergence and analytically known guarantees on parameter update rates, which makes them especially well suited for particle accelerator problems [26]. Such methods can be easily implemented via custom python scripts that read and write from machine components via network systems such as EPICS [27] and have been implemented in powerful optimization software such as OCELOT for online accelerator tuning [28]. The main benefit of adaptive methods is that they can be applied online in real time to drifting accelerator systems. For example, these methods have now been applied to automatically and quickly maximize the output power of FEL light at both the LCLS and the European XFEL and are able to compensate for un-modeled time variation in real time while optimizing 105 parameters simultaneously [29]. Another example of the benefit of these approaches is for multi-objective optimization which is typically done offline via extremely lengthy simulation studies. Adaptive methods have been demonstrated for real-time online multi-objective optimization of the electron beam line at AWAKE at CERN for simultaneous emittance growth minimization and trajectory control [30]. These adaptive methods have also been applied for online RL in which optimal feedback control policies were learned directly from data to learn optimal feedback control policies [31]. These methods have also been demonstrated at FACET to provide non-invasive LPS diagnostics that can actually predict and actively track time-varying TCAV measurements as both accelerator components and initial beam distributions drift with time [32].

1.1. Adaptive Machine Learning

The first adaptive ML approach combining ML and adaptive feedback for time-varying particle accelerator systems was recently developed for real time automatic control of the LPS of the LCLS electron beam [33]. The adaptive ML approach was demonstrated to combine the best of each family of tools: the power of ML tools such as NNs to learn complex relationships directly from measured data and the robust ability of adaptive feedback to handle time-varying noisy and unknown system dynamics. This adaptive ML approach has the potential to solve one of the main limitations in terms of matching the predictions of models (physics-based or surrogate) to actual accelerator beams, by adaptively identifying the initial beam distributions entering the accelerators [34], whose knowledge is needed for accurate model-based predictions.

In this work, our primary interest is in developing adaptive controls and diagnostics for time-varying conditions where it is possible that both the accelerator parameters and the initial beam entering the accelerator change as a function of time in unpredictable ways. Such uncertain time variation is one of the main challenges of accelerators and what makes it so difficult to run online models as real time non-invasive diagnostics because even the perfect model will not be predictive if it is not initialized with the correct parameter values or the correct initial beam distribution whose dynamic evolution is then simulated.

2. Unknown Time-varying Systems

Control in the presence of uncertainty and time-variation is extremely challenging even for linear systems. Linear systems are very popular because they are simple local approximations of much more complicated nonlinear dynamics and can be analytically solved. Consider a general n-dimensional nonlinear system

$$\begin{aligned}\dot{\mathbf{x}} &= \mathbf{f}(\mathbf{x},\mathbf{u}), \\ \underbrace{\begin{bmatrix}\dot{x}_1\\ \dot{x}_2\\ \vdots\\ \dot{x}_n\end{bmatrix}}_{\dot{\mathbf{x}}} &= \underbrace{\begin{bmatrix}f_1(\mathbf{x},\mathbf{u})\\ f_2(\mathbf{x},\mathbf{u})\\ \vdots\\ f_n(\mathbf{x},\mathbf{u})\end{bmatrix}}_{\mathbf{f}(\mathbf{x},\mathbf{u})}, \quad \mathbf{x} = \begin{bmatrix}x_1\\ x_2\\ \vdots\\ x_n\end{bmatrix}, \quad \mathbf{u} = \begin{bmatrix}u_1\\ u_2\\ \vdots\\ u_m\end{bmatrix}.\end{aligned} \tag{1}$$

Within a small enough neighborhood of any point $(\mathbf{x}_0,\mathbf{u}_0) \in \mathbb{R}^{n\times m}$, we can approximate the system (1), by a linear system, based on the Jacobian matrices of $\mathbf{f}(\mathbf{x},\mathbf{u})$. Via Taylor series expansion, for $\|(\mathbf{x},\mathbf{u})-(\mathbf{x}_0,\mathbf{u}_0)\| \ll 1$, we can approximate the $f_i(x,t)$ as:

$$\begin{aligned} f_i(\mathbf{x},\mathbf{u}) \approx\; & f_i(\mathbf{x}_0,\mathbf{u}_0) \\ & + \frac{\partial f_i(\mathbf{x}_0,\mathbf{u}_0)}{\partial x_1}(x_1 - x_{0,1}) + \cdots + \frac{\partial f_i(\mathbf{x}_0,\mathbf{u}_0)}{\partial x_n}(x_n - x_{0,n}) \\ & + \frac{\partial f_i(\mathbf{x}_0,\mathbf{u}_0)}{\partial u_1}(u_1 - u_{0,1}) + \cdots + \frac{\partial f_i(\mathbf{x}_0,\mathbf{u}_0)}{\partial u_m}(u_m - u_{0,m}) \end{aligned}$$

Thereby we approximate the nonlinear system

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x},\mathbf{u})$$

with

$$\begin{aligned} \underbrace{\begin{bmatrix}\dot{x}_1\\ \dot{x}_2\\ \vdots\\ \dot{x}_n\end{bmatrix}}_{\dot{\mathbf{x}}} \approx & \underbrace{\begin{bmatrix}\frac{\partial f_1}{\partial x_1} & \frac{\partial f_1}{\partial x_2} & \cdots & \frac{\partial f_1}{\partial x_n}\\ \frac{\partial f_2}{\partial x_1} & \frac{\partial f_2}{\partial x_2} & \cdots & \frac{\partial f_2}{\partial x_n}\\ \vdots & \vdots & \vdots & \vdots\\ \frac{\partial f_n}{\partial x_1} & \frac{\partial f_n}{\partial x_2} & \cdots & \frac{\partial f_n}{\partial x_n}\end{bmatrix}}_{\text{Jacobian with respect to } \mathbf{x}} \underbrace{\begin{bmatrix}x_1 - x_{0,1}\\ x_2 - x_{0,2}\\ \vdots\\ x_n - x_{0,n}\end{bmatrix}}_{\mathbf{x}-\mathbf{x}_0} \\ & + \underbrace{\begin{bmatrix}\frac{\partial f_1}{\partial u_1} & \frac{\partial f_1}{\partial u_2} & \cdots & \frac{\partial f_1}{\partial u_m}\\ \frac{\partial f_2}{\partial u_1} & \frac{\partial f_2}{\partial u_2} & \cdots & \frac{\partial f_2}{\partial u_m}\\ \vdots & \vdots & \vdots & \vdots\\ \frac{\partial f_n}{\partial u_1} & \frac{\partial f_n}{\partial u_2} & \cdots & \frac{\partial f_n}{\partial u_m}\end{bmatrix}}_{\text{Jacobian with respect to } \mathbf{u}} \underbrace{\begin{bmatrix}u_1 - u_{0,1}\\ u_2 - u_{0,2}\\ \vdots\\ u_n - u_{0,m}\end{bmatrix}}_{\mathbf{u}-\mathbf{u}_0} + \underbrace{\begin{bmatrix}f_1(\mathbf{x}_0,\mathbf{u}_0)\\ f_2(\mathbf{x}_0,\mathbf{u}_0)\\ \vdots\\ f_n(\mathbf{x}_0,\mathbf{u}_0)\end{bmatrix}}_{\mathbf{f}(\mathbf{x}_0,\mathbf{u}_0)}, \end{aligned}$$

with all of the above derivatives calculated at the point $(\mathbf{x}_0,\mathbf{u}_0)$. For any point $(\mathbf{x}_0,\mathbf{u}_0)$, we can always change coordinates to

$$(\mathbf{y},\mathbf{v}) = (\mathbf{x},\mathbf{u}) - (\mathbf{x}_0,\mathbf{u}_0),$$

which satisfy

$$\dot{\mathbf{y}} = \mathbf{f}(\mathbf{y}+\mathbf{x}_0,\mathbf{u}+\mathbf{u}_0) = \mathbf{f}_2(\mathbf{y},\mathbf{u}).$$

Therefore without loss of generality we can consider the case $(\mathbf{x}_0,\mathbf{u}_0) = (\mathbf{0},\mathbf{0})$, and from now on we ignore the term $\mathbf{f}(\mathbf{0},\mathbf{0})$, recognizing that it is just a constant disturbance. Next, if we define the constants

$$a_{ij} = \frac{\partial f_i}{\partial x_j}(\mathbf{0},\mathbf{0}), \quad b_{ij} = \frac{\partial f_i}{\partial u_j}(\mathbf{0},\mathbf{0}),$$

then we can write our system as

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \vdots \\ \dot{x}_n \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \vdots & \vdots & \vdots & \vdots \\ a_{n1} & a_{n2} & \dots & a_{nn} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} + \begin{bmatrix} b_{11} & b_{12} & \dots & b_{1m} \\ b_{21} & b_{22} & \dots & b_{2m} \\ \vdots & \vdots & \vdots & \vdots \\ b_{n1} & b_{n2} & \dots & b_{nm} \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \\ \vdots \\ u_m \end{bmatrix}$$

or

$$\dot{\mathbf{x}} = A\mathbf{x} + B\mathbf{u}. \tag{2}$$

If a linear feedback control of the form

$$\mathbf{u} = -K\mathbf{x} \tag{3}$$

is used, the resulting closed loop system is still linear

$$\dot{\mathbf{x}} = (A - BK)\mathbf{x}, \tag{4}$$

and can be analytically solved with the help of Laplace transforms to get

$$\mathbf{x}(t) = e^{(A-BK)t}\mathbf{x}(0) = e^{A_c t}\mathbf{x}(0), \quad A_c = A - BK. \tag{5}$$

Finally, the transient dynamics and stability of the system (4) are completely defined by the eigenvalues of the closed loop system matrix A_c because for any matrix, A_c, there exists an invertible matrix P such that

$$P^{-1}A_cP = J = \text{Block Diagonal}[J_1, J_2, \dots, J_r], \tag{6}$$

where J_i is a Jordan block associated with the eigenvalue λ_i of A_c, where Jordan blocks of order one are

$$J_i = \lambda_i,$$

and a Jordan blocks or order m_i are:

$$J_i = \begin{bmatrix} \lambda_i & 1 & 0 & \dots & \dots & 0 \\ 0 & \lambda_i & 1 & 0 & \dots & 0 \\ \vdots & 0 & \ddots & 0 & 0 & \vdots \\ \vdots & 0 & 0 & \ddots & 0 & 0 \\ \vdots & 0 & 0 & 0 & \lambda_i & 1 \\ 0 & 0 & 0 & 0 & 0 & \lambda_i \end{bmatrix}_{m_i \times m_i} .$$

Using the matrix exapnsion (6), we can rewrite the matrix exponential as

$$e^{A_c t} \quad = \quad e^{PJP^{-1}t} = Pe^{Jt}P^{-1}, \tag{7}$$

to get the solution

$$\mathbf{x}(t) = e^{(A-BK)t}\mathbf{x}(0) = e^{A_c t}\mathbf{x}(0) = Pe^{Jt}P^{-1}\mathbf{x}(0), \quad A_c = PJP^{-1}, \tag{8}$$

which, after some algebra, can be rewritten as

$$\mathbf{x}(t) = \left(\sum_{i=1}^{r} \sum_{k=1}^{m_i} t^{k-1} e^{\lambda_i t} R_{ik} \right) \mathbf{x}(0).$$

The eigenvalues of $A_c = A - BK$ have found their way into the exponential terms $e^{\lambda_i t}$. Expanding eigenvalues as a sum of real and imaginary parts, $\lambda_i = \lambda_{r,i} + i\lambda_{I,i}$, we can expand the exponential as

$$e^{\lambda_i t} = e^{(\lambda_{r,i} + i\lambda_{I,i})t} = e^{\lambda_{r,i} t} e^{i\lambda_{I,i} t} = e^{\lambda_{r,i} t}[\cos(\lambda_{I,i} t) + i \sin(\lambda_{I,i} t)].$$

Clearly the imaginary parts of the eigenvalues define the resonant frequencies of the system while the real parts control whether the trajectories of the dynamics converge or diverge exponentially. If we choose our feedback controller $-Kx$ such that the closed loop matrix $A - BK$, is Hurwitz, so its eigenvalues have negative real parts, then all $\lambda_{r,i} < 0$, and the system's trajectory exponentially converges to the origin, and the system is globally exponentially stable. This simple analysis of linear systems has lead to the popularity of simple proportional integral derivative (PID) control which is by far the most common form of feedback control.

If however, the original nonlinear system that we are controlling is time-varying

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, \mathbf{u}, t), \tag{9}$$

then by the same arguments as above, we can approximate our system (9) at any instant of time, within a small neighborhood of some point, by the linear time-varying system

$$\dot{\mathbf{x}} = A(t)\mathbf{x} + B(t)\mathbf{u}, \tag{10}$$

but that is where the similarity ends. For time-varying systems an eigenvalue analysis is completely useless (except in some very special cases, such as periodic or arbitrarily slowly varying systems) and determining whether systems (10) is stable or not and therefore designing stabilizing controllers, is incredibly difficult and requires a nonlinear Lyapunov analysis [35]. The uselessness of eigenvalues for time-varying systems is demonstrated by a few simple examples. Consider the system

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} -1 + 1.5\cos^2(t) & 1 - 1.5\sin(t)\cos(t) \\ -1 - 1.5\sin(t)\cos(t) & -1 + 1.5\sin^2(t) \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix},$$

which has constant eigenvalues with negative real parts $\lambda_{\pm} = -0.25 \pm 0.25\sqrt{7}i$, but which is exponentially unstable with solution

$$\begin{bmatrix} x_1(t) \\ x_2(t) \end{bmatrix} = \begin{bmatrix} e^{0.5t}\cos(t) & e^{-t}\sin(t) \\ -e^{0.5t}\sin(t) & e^{-t}\cos(t) \end{bmatrix} \begin{bmatrix} x_1(0) \\ x_2(0) \end{bmatrix}.$$

Another example is the linear time-varying system

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} -1 + 5\cos(t)\sin(t) & -5\cos^2(t) \\ 5\sin^2(t) & -1 - 5\cos(t)\sin(t) \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix},$$

which has constant negative real eigenvalues $\lambda_i = -1$ and is also unstable. The system

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} -\frac{11}{2} + \frac{15}{2}\sin(12t) & \frac{15}{2}\cos(12t) \\ \frac{15}{2}\cos(12t) & -\frac{11}{2} - \frac{15}{2}\sin(12t) \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix},$$

has constant eigenvalues, one of which is positive real, $\lambda_i = \{2, -13\}$, but is stable.

As seen from the examples above, even when we know everything about a system, if it is time-varying its stability properties are not obvious. On top of this a whole new level of difficulty is added when the system is also unknown. Consider an extremely simple 1D example of a one parameter system with dynamics

$$\dot{x} = x + b(t)u_1(x, t), \tag{11}$$

where $u_1(x, t)$ is a feedback controller and $b(t)$ is an unknown time-varying function such as $b(t) = b\cos(\omega t)$. The stabilization of such a system with unknown control direction,

$b(t)$, was an open problem in control theory for a long time. In 1985 a solution was proposed for the simple time-invariant case of $b(t) \equiv b$ [36,37], this solution suffered from an unbounded growing overshoot for time-varying $b(t)$ with hanging sign, eventually destabilizing and destroying any physical system. In 2012, problem (11) was finally solved with a novel model-independent approach [38,39], whose feedback forces the dynamics (11) to have an average behavior described by a new system of the form

$$\dot{\bar{x}} = \bar{x} - b^2(t)u_2(\bar{x}, t), \quad \|x(t) - \bar{x}(t)\| < \epsilon \tag{12}$$

with arbitrarily small $\epsilon > 0$, where the unknown control direction, $b(t)$, has become squared and can be stabilized automatically. Because of arbitrarily close proximity of the trajectories of (12) to (11), stabilization of (12) is equivalent to stabilizing (11).

The results in [38,39] actually solved the much more general problem of stabilizing a n-dimensional nonlinear time-varying unknown system of the form

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, t) + \mathbf{g}(\mathbf{x}, t)\mathbf{u}(\mathbf{x}, t),$$

where **f** and **g** are both nonlinear, time-varying, and analytically unknown. The method has now been generalized further with analytical proofs of stability for non-differentiable systems as well as systems not affine in control [40,41], of the form

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, \mathbf{u}, t),$$

and has been utilized in various particle accelerator applications [26,29–33].

3. Machine Learning for Time-Varying Systems

One application of machine learning is to try to learn unknown system relationships directly from data. This has been especially popular in the accelerator community for surrogate models and diagnostics based on them. Consider a family of N_p parameters in an accelerator, $\mathbf{p} = (p_1, \ldots, p_{N_p})$, which might include the currents of magnet power supplies and RF cavity amplitude and phase settings throughout an accelerator. Theoretically each set of parameter settings $\mathbf{p}_i$ maps, via some complicated nonlinear function, F, to an observable O_i according to

$$O_i = F(\mathbf{p}_i), \tag{13}$$

which may be, for example, the 2D (z, E) LPS distribution of the accelerator beam at a particular location. An ML tool can learn a close approximation $\hat{F}$ of the unknown function F directly from a data set D which contains enough pairs of parameter settings and their corresponding beam distributions

$$\hat{O}_i = \hat{F}(\mathbf{p}_i), \quad D = \{(\mathbf{p}_1, O_1), \ldots, (\mathbf{p}_m, O_m)\}, \quad e = \sum_{i=1}^{m} (\hat{O}_i - O_i)^2 \tag{14}$$

by minimizing some measure of error, such as e above, between ML predictions and observed data over the entire data set. Such a surrogate model approach has become popular recently for particle accelerator applications. For example, this has been done to map accelerator parameters to the LPS of the LCLS beam with the approximation $\hat{F}$ taking the form of a deep neural network [8].

It is known that both accelerator components and the initial beam distributions that are formed on photocathodes vary unpredictably with time and measurements are typically noisy and have arbitrary offsets, such as phase shifts of RF cables and analog components due to temperature and changing relationships between magnetic fields and power supplies due to hysteresis. Our goal is to tackle the problem of time varying systems in which (13) is replaced with

$$O_i(t) = F(\mathbf{p}_i(t), \boldsymbol{\rho}_0(t), t), \tag{15}$$

where we emphasize that the parameters $\mathbf{p}_i(t)$ drift with time, that the initial beam distribution $\boldsymbol{\rho}_0(t)$ entering any accelerator section drifts with time, and that the overall relationship between beam observables and parameters drifts with time (t). Furthermore, we recognize that at any observations of the accelerator and its beam are actually of the form $(\hat{\mathbf{p}}(t)_i, O_i)$ and $\hat{\boldsymbol{\rho}}_0(t)$, where $\hat{\mathbf{p}}_i(t)$ is an estimate of parameters $\mathbf{p}_i(t)$ and $\hat{\boldsymbol{\rho}}_0(t)$ an estimate of the uncertain initial beam distribution $\boldsymbol{\rho}_0(t)$. Therefore, a surrogate model such as a deep neural network will only result in an approximation that is valid for a small time interval, whose performance will drift as the accelerator's characteristics drift away from any collected data set.

Our approach is to implement adaptive ML which combines online simulations, adaptive feedback, and ML for systems of the form (15), which can be summarized as

$$\hat{O}_i(t) = \hat{F}(\hat{\mathbf{p}}_i(t), \hat{\boldsymbol{\rho}}_0(t), \mathbf{w}(\mathbf{t})), \qquad \hat{\boldsymbol{\rho}}_0(t) = \hat{F}_{\boldsymbol{\rho}}(\hat{\mathbf{p}}_i(t), \mathbf{w}_{\boldsymbol{\rho}}(\mathbf{t})), \tag{16}$$

$$\hat{\psi}(t) = M_{\psi}(\hat{\mathbf{p}}(t), \hat{\boldsymbol{\rho}}_0(t)), \qquad e_{\hat{\psi}}(t) = \int \left(\hat{\psi}(x,t) - \psi(x,t)\right)^2 dx, \tag{17}$$

where $\hat{F}$ represents an adaptive model-ML hybrid data-based learned representation of the observable O, where a trained surrogate model such as a convolutional neural network initially predicts the observable $\hat{O}$, with the prediction then refined by an online model. The adaptive ML model $\hat{F}$ receives time-varying parameter estimates $\hat{\mathbf{p}}(t)$ and an estimate of the beam's phase space $\hat{\boldsymbol{\rho}}_0(t)$ from a second model which is adaptively tuned based on the error $e_{\hat{\psi}}(t)$ quantifying the match between its prediction $\hat{\psi}(t)$ and the detected $\psi(t)$ value of a rich non-invasively measured beam characteristic such as the beam's energy spread spectrum. The parameter estimates, $\hat{\mathbf{p}}(t)$, ML weights $\mathbf{w}(t)$, and initial beam distribution $\hat{\boldsymbol{\rho}}(t)$ are all adaptively tuned utilizing a model-independent adaptive feedback control method.

4. Controls and Diagnostics for a 22 Dimensional System

To demonstrate some of the ideas described above, we perform a simulation-based study of an incredibly high-dimensional system in which we control 22 parameters which are the quadrupole magnets in the low energy beam transport of the Los Alamos Neutron Science Center (LANSCE) linear accelerator, as shown in Figure 1. The fact that our system has 22 dimensions makes it clear that ML-based studies are very challenging because even a very coarse grid search of 10 steps per parameter results in a staggering 10^{22} data points.

In this work we simulate beam dynamics according to the Kapchinsky-Vladimirsky (K-V) equations [42] which describe the evolution of the rms beam sizes $(X(z), Y(z))$ as

$$X''(z) = \frac{\epsilon_{rx}^2}{X^3(z)} + \frac{2K}{X(z)+Y(z)} - \Theta(z)X(z), \tag{18}$$

$$Y''(z) = \frac{\epsilon_{ry}^2}{Y^3(z)} + \frac{2K}{X(z)+Y(z)} + \Theta(z)Y(z), \tag{19}$$

$$\Theta(z) = \sum_{n=1}^{22} I_n(z)\Theta_n, \quad z \in [0, L = 11.7 \text{ m}] \tag{20}$$

where K is the beam perveance, a measure of how space-charge dominated the beam is, ϵ_{rx} and ϵ_{ry} are beam emittance in the x and y axis respective, $I_n(z)$ are indicator functions which are non-zero at the locations of the quadrupole magnets and Θ_n are the quarupole focusing strengths defined as

$$\Theta_n = \sqrt{\frac{|qG_n|}{mc\gamma\beta}}, \quad G_n = \frac{0.8\pi INv}{r^2}, \quad \beta = v/c, \quad \gamma = \frac{1}{\sqrt{1-\frac{v^2}{c^2}}} \tag{21}$$

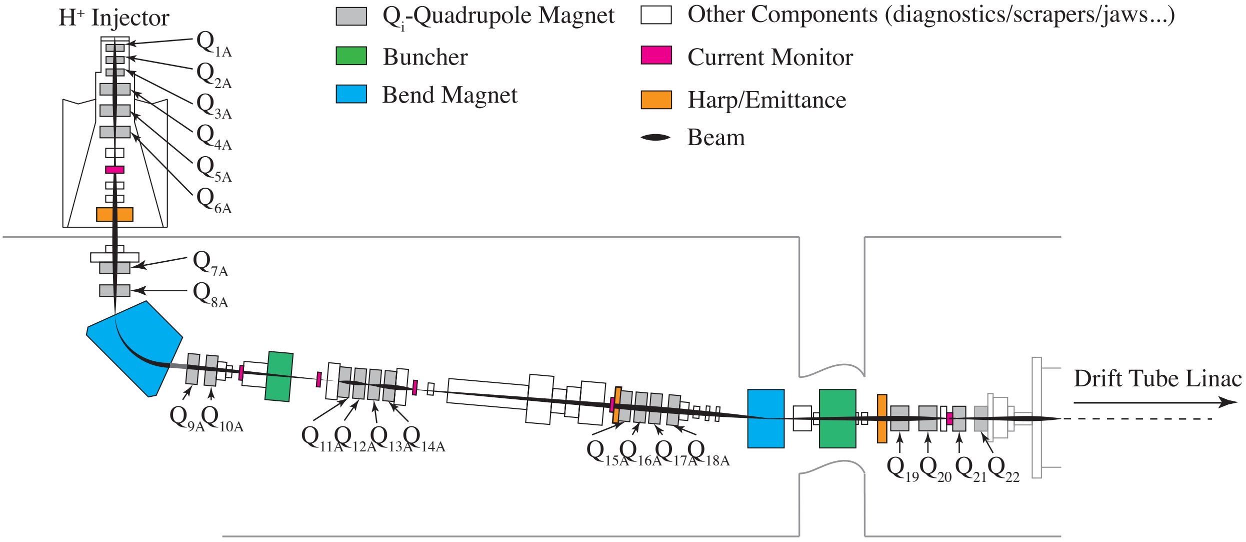

Figure 1. An overview of the LANSCE H$^+$ low energy beam transport and its 22 quadrupole magnets is shown.

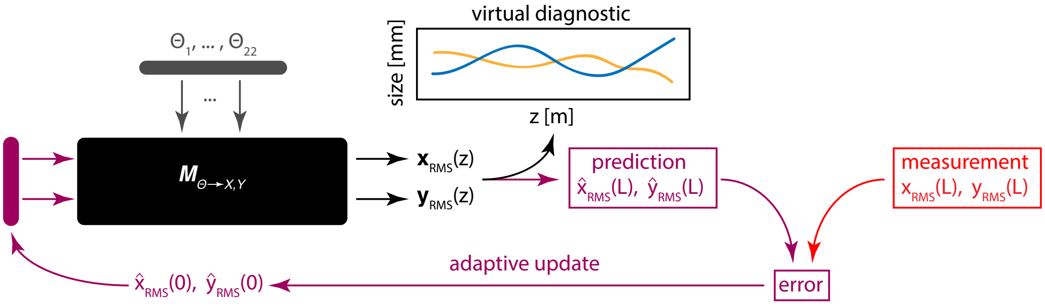

Figure 2. A high level view of the adaptive NN diagnostic $M_{\Theta \to X,Y}$ is shown.

doi:10.20944/preprints202103.0301.v1

where G_n is the magnetic field of quadrupole n in Gauss/cm, r is the magnet pole-tip radius, N is the number of turns per pole, ν is the efficiency factor and I is the magnet current.

In designing an adaptive diagnostic, one must make a choice about what parameter measurements are accurate and which might drift with time. In this case, we assume that we can accurately measure magnet settings and create an adaptive ML model for this system which serves as a non-invasive diagnostic by mapping magnet settings to beam profiles along the accelerator section, this NN-based model will be referred to as $M_{\Theta \to X,Y}$ and its output is two vectors of length 200 each, which represent estimates of $(X(z), Y(z))$ over the $z \in [0, L = 11.7 \text{ m}]$ range with 58 cm longitudinal resolution.

A typical surrogate model approach would simply run many iterations of the simulation (18)-(20) to learn a mapping from quadrupole settings to beam size along the LEBT. However, in our case, we know that the beam's initial size $(X(z=0), Y(z=0))$ coming out of the source will slowly drift unpredictably with time according to some unknown time-varying functions:

$$X(z=0,t) = F_X(t), \quad Y(z=0,t) = F_Y(t). \tag{22}$$

Such a drift is impossible to measure in real time during accelerator operations, it requires lengthy wire scanner-based measurements that interrupt operations. Taking this into account, although we do not expect to be able to measure $(X(z=0,t), Y(z=0,t))$, we still design our adaptive diagnostic with estimates $(\hat{X}(z=0,t), \hat{Y}(z=0,t))$ to be used as inputs. These estimate inputs will give our ML model the flexibility to be adaptive, to track changes in real time. In order to update our estimates, we need one final part which is some measurement of the beam that can be compared to the ML-based prediction. In this case the beam size can be estimated in real time at the end of the LEBT, $(X(z=L,t), Y(z=L,t))$ by looking at the amount of current lost on a pair of vertical and horizontal slits. The plan is then to compare the model's prediction of the final horizontal and vertical beam sizes to their measurements and calculate an error

$$e(t) = |\hat{X}(z=L,t) - X(z=L,t)| + |\hat{Y}(z=L,t) - Y(z=L,t)|, \tag{23}$$

and use that error to adaptively tune the input beam size estimates by using the methods in [40,41] to track the time-varying input beam distribution:

$$\frac{d\hat{X}(z=0,t)}{dt} = \sqrt{\alpha\omega}\cos(\omega t + ke(t)), \quad \frac{d\hat{Y}(z=0,t)}{dt} = \sqrt{\alpha\omega}\sin(\omega t + ke(t)). \tag{24}$$

A high level overview of this adpative ML diagnostic approach is shown in Figure 2, where the NN $M_{\Theta \to X,Y}$ has dense connected layers with relu activation functions.

To train the model we simulated the dynamics (18)-(20) 600 thousand times and used 590 thousand of the generated trajectories and their corresponding quadrupole magnet values for training, reserving 10 thousand for model validation. Each simulation generated random initial conditions $(X(0), Y(0)$ by uniformly sampling distributions with mean values of 2 mm and 1.75 mm respectively over a range of ±0.5 mm, which is realistic based on experimental data. The quadrupole magnet values were also sampled from uniform distributions centered on a known good setup over a range of ± 1.

Considering the root-mean-square error (RMSE) as the metric of accuracy for both the X and Y trajectories,

$$\text{RMSE}_x = \sqrt{\sum_{n=1}^{200} \frac{\left(\hat{X}(n) - X(n)\right)^2}{200}}, \quad \text{RMSE}_y = \sqrt{\sum_{n=1}^{200} \frac{\left(\hat{Y}(n) - Y(n)\right)^2}{200}}, \tag{25}$$

a histogram of the sum of these errors is shown in Figure 3 for the 10k validation set as well as the worst and best predictions from this set. In Figure 3 we also show the sum of these

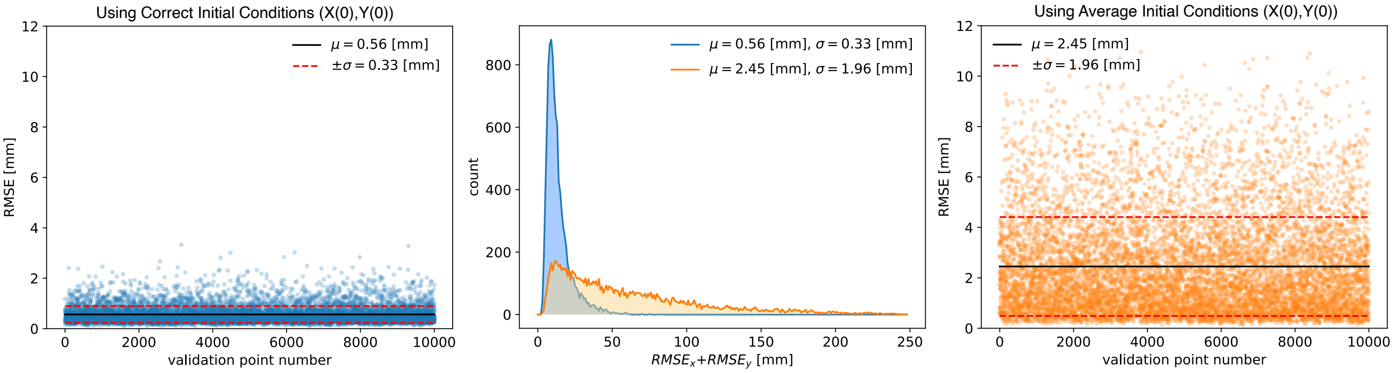

Figure 3. Comparison of NN predictions when using the correct initial conditions and when using the mean value.

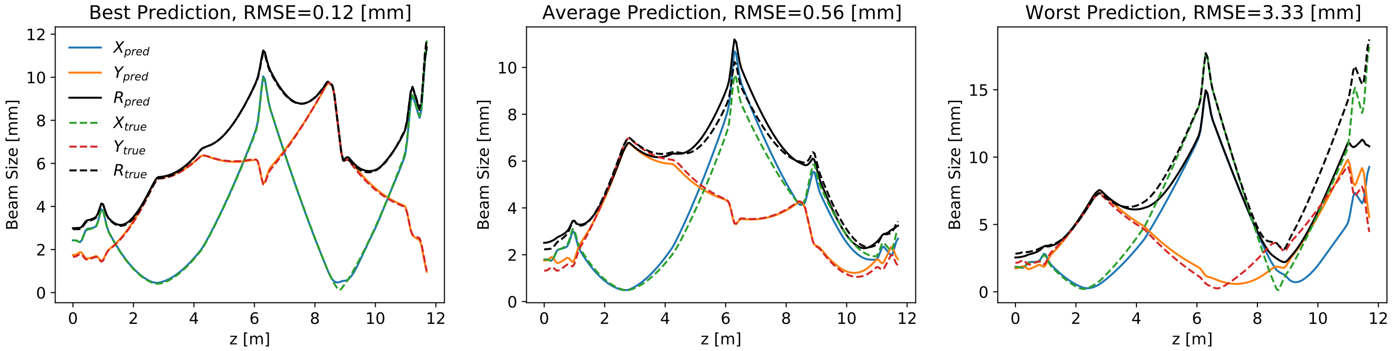

Figure 4. Comparison of the best (left), average (midddle) and worst (right) NN predictions for beam envelopes.

errors for the validation set if the input beam sizes to the NN are simply set as the average values of 2 mm and 1.75 mm. Figure 4 shows the beam envelopes for the best, average, and worst predictions of the NN when using the correct initial conditions as input.

Finally, in Figure 5 we show the results of using this adaptive ML-based diagnostic to track time varying initial beam sizes.

By a similar principal, it is possible to design a second model which can be used to guide feedback control for accelerator tuning, by solving the inverse problem of mapping beam sizes to the quadrupole values that are required to achieve them, referred to as $M_{X,Y\rightarrow\Theta}$. A high level view of such an adaptive control approach is shown in Figure 6. Such a model would require a diagnostic for comparing achieved and desired beam profiles and this type of approach was first demonstrated on the LCLS for automatic control of the LPS whose measurement was available in real time via a transverse deflecting cavity (TCAV) [33].

5. Conclusions

We have shown how combining adaptive model-independent methods with machine learning tools for adaptive ML combines the robustness of adaptive methods to time variation and uncertainty, with the global and extremely fast predictions of complex nonlinear dynamics. The main trick is to introduce extra inputs which may be tuned to modify the ML structure's predictions based on some measurement. These inputs do not have to be physically measurable quantities, but they should have a physically meaningful effect on the predictions.

Funding: This research was supported by Los Alamos National Laboratory's Laboratory-Directed Research and Development (LDRD) program by project number 20210595DR.

Data Availability Statement: The data that support the findings of this study are available from the corresponding author upon reasonable request.

Conflicts of Interest: “The authors declare no conflict of interest.”

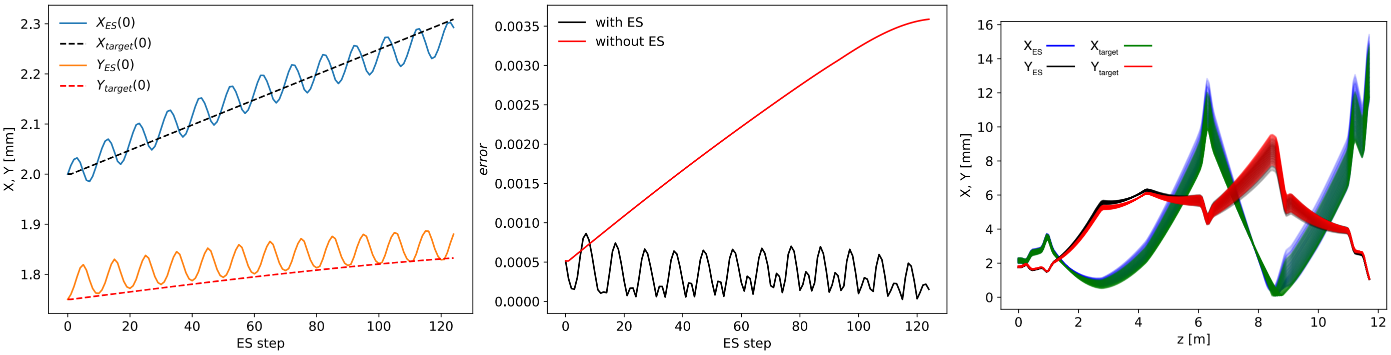

Figure 5. The adaptive diagnostic is able to track the time-varying beam in real time by tuning the input beam sizes based only on the error measured at the end of the beamline. The left most subplot shows the changing initial beam sizes ad how they are tracked. The middle subplot shows the growth of the error without adaptive tuning and with ES. The right most plot shows all of the trajectories overlapped as they are tracked by the adaptive diagnostic.

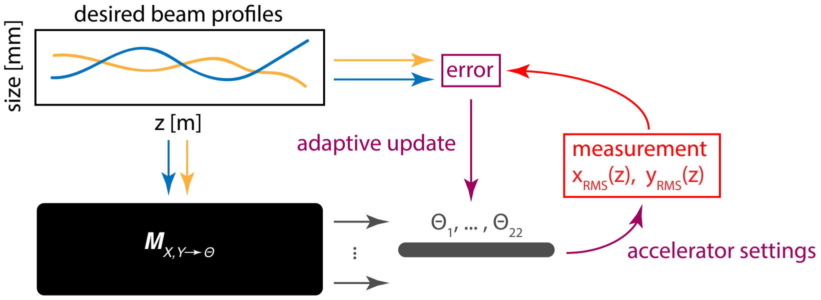

Figure 6. A high level view of the adaptive NN controller $M_{X,Y\to\Theta}$ is shown.

1. Friedman, J.; Hastie, T.; Tibshirani, R. *The elements of statistical learning*; Vol. 1, Springer series in statistics New York, 2001.
2. Kleene, S.C. Representation of events in nerve nets and finite automata. Technical report, RAND PROJECT AIR FORCE SANTA MONICA CA, 1951.
3. Rasmussen, C.E. Gaussian processes in machine learning. Summer school on machine learning. Springer, 2003, pp. 63–71.
4. Sutton, R.S.; Barto, A.G. *Reinforcement learning: An introduction*; MIT press, 2018.
5. Li, Y.; Cheng, W.; Yu, L.H.; Rainer, R. Genetic algorithm enhanced by machine learning in dynamic aperture optimization. *Physical Review Accelerators and Beams* **2018**, *21*, 054601.
6. Edelen, A.; Neveu, N.; Frey, M.; Huber, Y.; Mayes, C.; Adelmann, A. Machine learning for orders of magnitude speedup in multiobjective optimization of particle accelerator systems. *Physical Review Accelerators and Beams* **2020**, *23*, 044601.
7. Kranjvcević, M.; Riemann, B.; Adelmann, A.; Streun, A. Multiobjective optimization of the dynamic aperture using surrogate models based on artificial neural networks. *Physical Review Accelerators and Beams* **2021**, *24*, 014601.
8. Emma, C.; Edelen, A.; Hogan, M.; O'Shea, B.; White, G.; Yakimenko, V. Machine learning-based longitudinal phase space prediction of particle accelerators. *Physical Review Accelerators and Beams* **2018**, *21*, 112802.
9. Hanuka, A.; Emma, C.; Maxwell, T.; Fisher, A.S.; Jacobson, B.; Hogan, M.; Huang, Z. Accurate and confident prediction of electron beam longitudinal properties using spectral virtual diagnostics. *Scientific Reports* **2021**, *11*, 1–10.
10. Zhu, J.; Chen, Y.; Brinker, F.; Decking, W.; Tomin, S.; Schlarb, H. Deep Learning-Based Autoencoder for Data-Driven Modeling of an RF Photoinjector. *arXiv preprint arXiv:2101.10437* **2021**.
11. McIntire, M.; Cope, T.; Ermon, S.; Ratner, D.; others. Bayesian optimization of FEL performance at LCLS. Proceedings of the 7th International Particle Accelerator Conference, 2016.
12. Li, Y.; Rainer, R.; Cheng, W. Bayesian approach for linear optics correction. *Physical Review Accelerators and Beams* **2019**, *22*, 012804.

13. Hao, Y.; Li, Y.; Balcewicz, M.; Neufcourt, L.; Cheng, W. Reconstruction of Storage Ring's Linear Optics with Bayesian Inference. *arXiv preprint arXiv:1902.11157* **2019**.
14. Duris, J.; Kennedy, D.; Hanuka, A.; Shtalenkova, J.; Edelen, A.; Baxevanis, P.; Egger, A.; Cope, T.; McIntire, M.; Ermon, S.; others. Bayesian optimization of a free-electron laser. *Physical review letters* **2020**, *124*, 124801.
15. Li, Y.; Hao, Y.; Cheng, W.; Rainer, R. Analysis of beam position monitor requirements with Bayesian Gaussian regression. *arXiv preprint arXiv:1904.05683* **2019**.
16. Shalloo, R.; Dann, S.; Gruse, J.N.; Underwood, C.; Antoine, A.; Arran, C.; Backhouse, M.; Baird, C.; Balcazar, M.; Bourgeois, N.; others. Automation and control of laser wakefield accelerators using Bayesian optimization. *Nature communications* **2020**, *11*, 1–8.
17. Fol, E.; de Portugal, J.C.; Franchetti, G.; Tomás, R. Optics corrections using machine learning in the LHC. Proceedings of the 2019 International Particle Accelerator Conference, Melbourne, Australia, 2019.
18. Fol, E.; de Portugal, J.C.; Tomás, R.; others. Unsupervised Machine Learning for Detection of Faulty Beam Position Monitors. Proc. 10th Int. Particle Accelerator Conf.(IPAC'19), Melbourne, Australia, 2019, Vol. 2668.
19. Arpaia, P.; Azzopardi, G.; Blanc, F.; Bregliozzi, G.; Buffat, X.; Coyle, L.; Fol, E.; Giordano, F.; Giovannozzi, M.; Pieloni, T.; others. Machine learning for beam dynamics studies at the CERN Large Hadron Collider. *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment* **2021**, *985*, 164652.
20. Adelmann, A. On nonintrusive uncertainty quantification and surrogate model construction in particle accelerator modeling. *SIAM/ASA Journal on Uncertainty Quantification* **2019**, *7*, 383–416.
21. Hirlaender, S.; Bruchon, N. Model-free and Bayesian Ensembling Model-based Deep Reinforcement Learning for Particle Accelerator Control Demonstrated on the FERMI FEL. *arXiv preprint arXiv:2012.09737* **2020**.
22. Kain, V.; Hirlander, S.; Goddard, B.; Velotti, F.M.; Della Porta, G.Z.; Bruchon, N.; Valentino, G. Sample-efficient reinforcement learning for CERN accelerator control. *Physical Review Accelerators and Beams* **2020**, *23*, 124801.
23. Bruchon, N.; Fenu, G.; Gaio, G.; Lonza, M.; O'Shea, F.H.; Pellegrino, F.A.; Salvato, E. Basic reinforcement learning techniques to control the intensity of a seeded free-electron laser. *Electronics* **2020**, *9*, 781.
24. Bruchon, N.; Fenu, G.; Gaio, G.; Lonza, M.; Pellegrino, F.A.; Salvato, E. Toward the application of reinforcement learning to the intensity control of a seeded free-electron laser. 2019 23rd International Conference on Mechatronics Technology (ICMT). IEEE, 2019, pp. 1–6.
25. O'Shea, F.; Bruchon, N.; Gaio, G. Policy gradient methods for free-electron laser and terahertz source optimization and stabilization at the FERMI free-electron laser at Elettra. *Physical Review Accelerators and Beams* **2020**, *23*, 122802.
26. Scheinker, A.; Krstić, M. *Model-free stabilization by extremum seeking*; Springer, 2017.
27. Dalesio, L.R.; Kozubal, A.; Kraimer, M. EPICS architecture. Technical report, Los Alamos National Lab., NM (United States), 1991.
28. Agapov, I.; Geloni, G.; Tomin, S.; Zagorodnov, I. OCELOT: A software framework for synchrotron light source and FEL studies. *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment* **2014**, *768*, 151–156.
29. Scheinker, A.; Bohler, D.; Tomin, S.; Kammering, R.; Zagorodnov, I.; Schlarb, H.; Scholz, M.; Beutner, B.; Decking, W. Model-independent tuning for maximizing free electron laser pulse energy. *Physical Review Accelerators and Beams* **2019**, *22*, 082802.
30. Scheinker, A.; Hirlaender, S.; Velotti, F.M.; Gessner, S.; Della Porta, G.Z.; Kain, V.; Goddard, B.; Ramjiawan, R. Online multi-objective particle accelerator optimization of the AWAKE electron beam line for simultaneous emittance and orbit control. *AIP Advances* **2020**, *10*, 055320.
31. Scheinker, A.; Scheinker, D. Extremum seeking for optimal control problems with unknown time-varying systems and unknown objective functions. *International Journal of Adaptive Control and Signal Processing* **2020**.
32. Scheinker, A.; Gessner, S. Adaptive method for electron bunch profile prediction. *Physical Review Special Topics-Accelerators and Beams* **2015**, *18*, 102801.
33. Scheinker, A.; Edelen, A.; Bohler, D.; Emma, C.; Lutman, A. Demonstration of model-independent control of the longitudinal phase space of electron beams in the linac-coherent light source with femtosecond resolution. *Physical review letters* **2018**, *121*, 044801.

34. Scheinker, A.; Cropp, F.; Paiagua, S.; Filippetto, D. Demonstration of adaptive machine learning-based distribution tracking on a compact accelerator: Towards enabling model-based 6D non-invasive beam diagnostics. *arXiv preprint arXiv:2102.10510* **2021**.
35. Khalil, H.K.; Grizzle, J.W. *Nonlinear systems*; Vol. 3, Prentice hall Upper Saddle River, NJ, 2002.
36. Mudgett, D.; Morse, A. Adaptive stabilization of linear systems with unknown high-frequency gains. *IEEE Transactions on Automatic Control* **1985**, *30*, 549–554.
37. Nussbaum, R.D. Some remarks on a conjecture in parameter adaptive control. *Systems & control letters* **1983**, *3*, 243–246.
38. Scheinker, A. Extremum seeking for stabilization. PhD thesis, UC San Diego, 2012.
39. Scheinker, A.; Krstić, M. Minimum-seeking for CLFs: Universal semiglobally stabilizing feedback under unknown control directions. *IEEE Transactions on Automatic Control* **2012**, *58*, 1107–1122.
40. Scheinker, A.; Scheinker, D. Bounded extremum seeking with discontinuous dithers. *Automatica* **2016**, *69*, 250–257.
41. Scheinker, A.; Scheinker, D. Constrained extremum seeking stabilization of systems not affine in control. *International Journal of Robust and Nonlinear Control* **2018**, *28*, 568–581.
42. Kapchinskij, I.; Vladimirskij, V. Limitations of proton beam current in a strong focusing linear accelerator associated with the beam space charge. Proceedings of the International Conference on High Energy Accelerators and Instrumentation. CERN Scientific Information Service Geneva, 1959, p. 274.