



## Model-independent particle accelerator tuning

Alexander Scheinker,<sup>\*</sup> Xiaoying Pang,<sup>†</sup> and Larry Rybarczyk<sup>‡</sup>

*Los Alamos National Laboratory, Los Alamos, New Mexico 87545, USA*

(Received 25 July 2013; published 21 October 2013)

We present a new model-independent dynamic feedback technique, rotation rate tuning, for automatically and simultaneously tuning coupled components of uncertain, complex systems. The main advantages of the method are: (1) it has the ability to handle unknown, time-varying systems, (2) it gives known bounds on parameter update rates, (3) we give an analytic proof of its convergence and its stability, and (4) it has a simple digital implementation through a control system such as the experimental physics and industrial control system (EPICS). Because this technique is model independent it may be useful as a real-time, in-hardware, feedback-based optimization scheme for uncertain and time-varying systems. In particular, it is robust enough to handle uncertainty due to coupling, thermal cycling, misalignments, and manufacturing imperfections. As a result, it may be used as a fine-tuning supplement for existing accelerator tuning/control schemes. We present multiparticle simulation results demonstrating the scheme's ability to simultaneously adaptively adjust the set points of 22 quadrupole magnets and two rf buncher cavities in the Los Alamos Neutron Science Center (LANSCE) Linear Accelerator's transport region, while the beam properties and rf phase shift are continuously varying. The tuning is based only on beam current readings, without knowledge of particle dynamics. We also present an outline of how to implement this general scheme in software for optimization, and in hardware for feedback-based control/tuning, for a wide range of systems.

DOI: [10.1103/PhysRevSTAB.16.102803](https://doi.org/10.1103/PhysRevSTAB.16.102803)

PACS numbers: 41.85.Lc, 02.30.Yy, 29.20.-c, 02.60.-x

### I. INTRODUCTION

#### A. Motivation

It is rarely possible to build exact, deterministic input to output models for complex physical systems such as particle accelerators. It is especially difficult when the behavior of the system is influenced by many coupled parameters. Since accelerators have many coupled parameters, they are prime candidates for genetic algorithm (GA) and multiobjective genetic algorithm (MOGA) based multidimensional, nonlinear optimization schemes. In fact, MOGAs and GAs have been used to successfully optimize many aspects of particle accelerators, such as magnet and radio frequency (rf) cavity design [1], photoinjector design [2], damping ring design [3], storage ring dynamics [4], global optimization of a lattice [5], neutrino factory design [6], simultaneous optimization of beam emittance and dynamic aperture [7], and free electron laser linac drivers [8]. A thorough review of GA for accelerator physics applications is given in [9].

After an accelerator design has been finalized and the accelerator has been constructed, one often encounters

time-varying and nonlinear coupling effects between the imperfectly manufactured and misaligned/unknown orientation components of the accelerator. In theory, accelerator design takes a certain level of uncertainty into account. In practice however, most accelerators require postmanufacture and postinstallation tuning. This is especially the case for facilities with limited real-time diagnostics and noise measurement. In this case components may have to be retuned after each shutdown or change in operating conditions. Effects such as unknown hysteresis curves and time-varying component thermal cycling also add to system uncertainty. A particular problem faced by many accelerator systems is the arbitrary phase shift of the rf systems, a time-varying uncertainty, requiring time consuming tuning such as phase scans. The method presented here is demonstrated to automatically adapt for time-varying properties, such as phase shift. A combination of the global optimization abilities of GAs, with a local, model-independent feedback technique such as the one presented here, has the potential to improve accelerator design and performance.

#### B. Results of the paper

In this work, we present a simple, model-independent technique, which can aid in parameter tuning because it does not, by design, assume any particular system model for optimizing/tuning and therefore may be implemented in hardware to automatically fine-tune multiple parameters and help mitigate unmodeled disturbances and component imperfections.

<sup>\*</sup>ascheink@lanl.gov

<sup>†</sup>xpang@lanl.gov

<sup>‡</sup>lrybarcyk@lanl.gov

*Published by the American Physical Society under the terms of the [Creative Commons Attribution 3.0 License](https://creativecommons.org/licenses/by/3.0/). Further distribution of this work must maintain attribution to the author(s) and the published article's title, journal citation, and DOI.*

For implementation, the user first defines a measurable cost function,  $C$ , to be minimized, whose analytic form may be unknown, such as the total particle loss along the length of a particle accelerator. The components  $p_i$  of the vector  $\mathbf{p} = (p_1, \dots, p_m)$  are parameters by which the cost may be influenced, such as the power source current settings feeding the quadrupole magnets in the accelerator lattice. The rotation rate (RR) tuning law is

$$p_i(n+1) = p_i(n) + \Delta \sqrt{\alpha \omega_i} \cos\{\omega_i n \Delta + kC[\mathbf{p}(n)]\}. \quad (1)$$

Initial settings  $\mathbf{p}(1)$  are chosen as usual, based on a physics model and a MOGA or other numerical optimization technique. The initial cost,  $C[\mathbf{p}(1)]$ , is calculated after the first run and new parameter values  $\mathbf{p}(2)$  are set according to (1). The waiting time between implementation  $n$  and  $n+1$  is chosen depending on component response/settling rates, data acquisition rates, and the rate of time variation of system components due to disturbance. Intuitively,  $\alpha$  is the magnitude of a high frequency ( $\omega_i$ ) dither being introduced into the system's dynamics,  $k$  is the gain of the controller/tuning algorithm, and  $\Delta$  is a time interval for digital implementation. The choices for the values of  $\omega_i$ ,  $\Delta$ ,  $k$ , and  $\alpha$  are discussed in detail in Sec. III.

We demonstrate, through a multiparticle simulation of the Los Alamos Neutron Science Center (LANSCE) low energy beam transport region, RR's ability to handle uncertainty by tuning up a 22 quadrupole lattice, and the phases of two rf buncher cavities, by minimizing a cost which is based on beam current loss along the transport region and the first two tanks of the drift tube linac, for a time-varying beam and time-varying rf phase drifts.

The RR approach may also be used as a simple numerical optimizer, in which adding new parameters  $p_{m+1}, p_{m+2}, \dots$  to (1) does not add significantly to computation time. Multiobjective optimization is implemented by replacing a single cost,  $C$ , with a combination  $C = C_1 + \dots + C_n$ , for any number of costs, such as  $C_1$  being total beam loss,  $C_2$  being total transverse beam size, etc. Also, different parameters may be updated based on different rates and costs (as in Sec. IVA), where components having different sensitivities may require different values of  $k_i$  and  $\alpha_i$ . For example, we may implement

$$p_i(n+1) = p_i(n) + \Delta \sqrt{\alpha_i \omega_i} \cos\{\omega_i \Delta n + k_i C_i[\mathbf{p}(n)]\}, \quad (2)$$

for notational simplicity, we stick to single values of  $k$ ,  $\alpha$ , and  $C$  throughout the analysis performed, which is applicable in the same manner to the above, more general scheme (2). Imposing restrictions on the parameters is straightforward and implemented as described in Sec. III.

As shown in the analysis in Sec. IIB, the scheme (1) is chosen so that, with proper choices of  $\omega$ ,  $k$ ,  $\alpha$ , and  $\Delta$ , (1) is the finite difference approximation of

$$\dot{p}_i = \sqrt{\alpha \omega_i} \cos\{\omega_i t + kC[\mathbf{p}(t), t]\}, \quad (3)$$

which on average follows the same trajectory as the system

$$\dot{\bar{p}}_i = -\frac{k\alpha}{2} \frac{\partial C[\bar{\mathbf{p}}(t), t]}{\partial \bar{p}_i}, \quad (4)$$

where the convention used here and throughout the remainder of the paper is  $\dot{\mathbf{p}} = \frac{\partial \mathbf{p}}{\partial t}$ .

RR is related to dithering-based optimization/stabilization schemes, known in control theory as extremum seeking (ES), which have been used for optimizing unknown outputs of known, stable systems, by tuning known controllers. Originally introduced in 1922 [10], an overview of its development is available in [11]. Recently, ES has been extended to perform stabilization and optimization of unknown, possible unstable systems [12]. RR is a further improvement and modification of those results [13].

### C. Optimization schemes

There are many existing model dependent numerical methods for multidimensional/multiparameter optimization, such as GA, MOGA, Newton-Raphson and gradient descent based on the analytic form of  $\nabla C$ . Many optimization methods are actually built into existing accelerator design codes [14].

The main strength of RR is its model-independent nature and ability to deal with multiple parameters simultaneously, even for time-varying systems, such as thermal cycling, or unexpected component damage. Some very simple, but computationally intensive and inefficient, model-independent methods are grid and random point searches, especially for systems with many parameters. Gradient descent, based on a numerical approximation of an unknown  $\nabla C$  is another model-independent approach, but especially in the case of a multiparameter, noisy calculation of  $C$ , may face difficulties, whereas RR is both robust to noise in  $C$  and does not need to try and estimate  $\nabla C$ . Only samples  $C(n)$  are required.

Simplex fitting, in the sense that it samples many different directions in multidimensional parameter space, has the most in common with RR. A major benefit of RR is that its complexity does not grow with parameter number, regardless of the number of parameters being tuned. The scheme basically depends on three choices, the values of  $k$ ,  $\omega$ , and  $\alpha$ . Regarding noisy data, the RR scheme is, on average, not influenced by noise, unless it happens to both match an RR parameter's perturbation frequency and be large in magnitude relative to that perturbation. Also, noise is easily handled by standard methods, such as averaging and filtering.

### D. Limitations

Although RR is model independent and able to tune many parameters simultaneously, unlike GA, it is a local technique, similar to gradient descent, and may become trapped in local minimums. Therefore, we plan on exploring (in future work) a combination of GA and RR, in which a GA is first used for global optimization followed by RR for local, in-hardware tuning, to make up for modeling errors and time variation.

## E. Organization

In Sec. II we give an explanation for the choice of the update scheme (1). In Sec. III we explain how to choose all RR parameters and describe digital implementation for general systems. In Sec. IV, we demonstrate the scheme's ability to handle uncertainty by tuning a 22 quadrupole lattice as well as two rf buncher cavities, in a simulation of the LANSCE accelerator transport region and first two tanks of the drift tube linac, for a varying input beam and drifting phase shifts. Finally, in Sec. V we provide background regarding RR analysis.

## II. TUNING METHOD

### A. Physical motivation

It is well known that by adding a fast, small oscillation into a system's dynamics, unexpected stability properties may be achieved. The classic example is of the inverted pendulum, whose vertical equilibrium point may be stabilized by rapidly vertically oscillating the pendulum's pivot point. The dynamics of this process were first analytically described in the 1950s by Kapitza [15]. The RR scheme has some similarities to this approach, in that we introduce high frequency oscillations into a system in order to force certain points of the state space to become stable equilibrium points towards which the system's trajectory converges. By abstracting this to a general state space and choosing such a point to be the minimum of a cost function, we are able to tune a wide range of systems towards various performance goals.

We start with a simple example, we do not introduce any destabilizing terms in (5) and (6), which are discussed in remark 1. To give a simple 2D overview of this method, we consider finding the minimum of a measurable function  $C(x, y)$ , for which we cannot simply implement a gradient descent for the trajectory of  $[x(t), y(t)]$  because we are unaware of its analytic form. We propose the following adaptive scheme:

$$\frac{\partial x}{\partial t} = \sqrt{\alpha\omega} \cos[\omega t + kC(x, y)] \quad (5)$$

$$\frac{\partial y}{\partial t} = \sqrt{\alpha\omega} \sin[\omega t + kC(x, y)]. \quad (6)$$

Note that although  $C(x, y)$  enters the argument of the adaptive scheme, we do not rely on any knowledge of the analytic form of  $C(x, y)$ , we simply assume that its value is available for measurement at different locations  $(x, y)$ .

The velocity vector,

$$\mathbf{v} = \left( \frac{\partial x}{\partial t}, \frac{\partial y}{\partial t} \right) = \sqrt{\alpha\omega} \{ \cos[\theta(t)], \sin[\theta(t)] \}, \quad (7)$$

where  $\theta(t) = \omega t + kC[x(t), y(t)]$ , has constant magnitude,  $\|\mathbf{v}\| = \sqrt{\alpha\omega}$ , and therefore the trajectory  $[x(t), y(t)]$  moves at a constant speed. However, the rate at which the direction of the trajectories' heading changes is a function of  $\omega$ ,  $k$ , and  $C[x(t), y(t)]$  expressed as

$$\frac{\partial \theta}{\partial t} = \omega + k \left( \frac{\partial C}{\partial x} \frac{\partial x}{\partial t} + \frac{\partial C}{\partial y} \frac{\partial y}{\partial t} \right). \quad (8)$$

Therefore, when the trajectory is heading in the correct direction, towards a decreasing value of  $C[x(t), y(t)]$ , the term  $k \frac{\partial C}{\partial r}$  is negative so the overall turning rate  $\frac{\partial \theta}{\partial t}$  (8) is decreased. On the other hand, when the trajectory is heading in the wrong direction, towards an increasing value of  $C[x(t), y(t)]$ , the term  $k \frac{\partial C}{\partial r}$  is positive, and the turning rate is increased. On average, the system ends up approaching the minimizing location of  $C[x(t), y(t)]$  because it spends more time moving towards it than away.

The ability of this direction-dependent turning rate scheme is apparent in the simulation of system (5) and (6), in Fig. 1. The system, starting at initial location  $x(0) = 1, y(0) = -1$ , is simulated for five seconds with update parameters  $\omega = 50, k = 5, \alpha = 0.5$ , and  $C(x, y) = x^2 + y^2$ . We compare the actual system's (5) and (6) dynamics with those of a system performing gradient descent:

$$\frac{\partial \bar{x}}{\partial t} \approx -\frac{k\alpha}{2} \frac{\partial C(\bar{x}, \bar{y})}{\partial \bar{x}} = -k\alpha \bar{x}, \quad (9)$$

$$\frac{\partial \bar{y}}{\partial t} \approx -\frac{k\alpha}{2} \frac{\partial C(\bar{x}, \bar{y})}{\partial \bar{y}} = -k\alpha \bar{y}, \quad (10)$$

whose behavior our system mimics on average, with the difference

$$\max_{t \in [0, T]} \|[x(t), y(t)] - [\bar{x}(t), \bar{y}(t)]\| \quad (11)$$

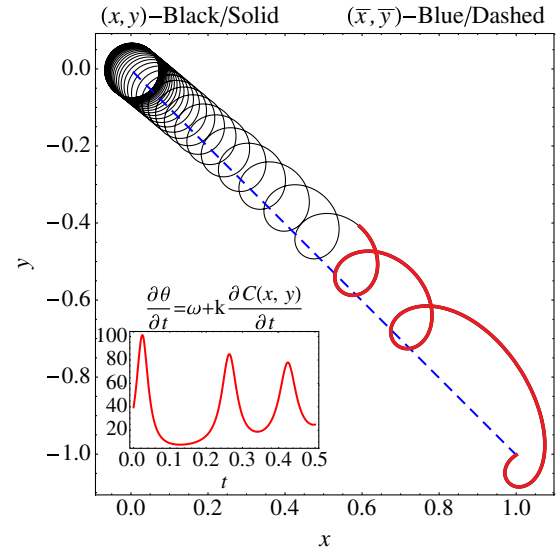


FIG. 1. The subfigure in the bottom left shows the rotation rate,  $\frac{\partial \theta}{\partial t} = \omega + k \frac{\partial C(x, y)}{\partial r}$ , for the part of the trajectory that is bold red, which takes place during the first 0.5 seconds of simulation. The rotation of the parameters' velocity vector  $\mathbf{v}(t)$  slows down when heading towards the minimum of  $C(x, y) = x^2 + y^2$ , at which time  $k \frac{\partial C}{\partial r} < 0$ , and speeds up when heading away from the minimum, when  $k \frac{\partial C}{\partial r} > 0$ . The system ends up spending more time heading towards and approaches the minimum of  $C(x, y)$ .

made arbitrarily small for any value of  $T$ , by choosing arbitrarily large values of  $\omega$ . The derivation of this relationship and of the rate of the gradient descent are given in Sec. V.

Towards the end of the simulation, when the system's trajectory is near the origin,  $C(x, y) \approx 0$ , and the dynamics of (5) and (6) are approximately

$$\frac{\partial x}{\partial t} \approx \sqrt{\alpha\omega} \cos(\omega t) \Rightarrow x(t) \approx \sqrt{\frac{\alpha}{\omega}} \sin(\omega t), \quad (12)$$

$$\frac{\partial y}{\partial t} \approx \sqrt{\alpha\omega} \sin(\omega t) \Rightarrow y(t) \approx -\sqrt{\frac{\alpha}{\omega}} \cos(\omega t), \quad (13)$$

a circle of radius  $\sqrt{\frac{\alpha}{\omega}}$ , which is made arbitrarily small by choosing small values of  $\alpha$  or large values of  $\omega$ . A detailed overview of how to choose the values  $k$ ,  $\alpha$ , and  $\omega$  is given in Sec. III. Convergence towards a maximum, rather than a minimum, is achieved by replacing  $k$  with  $-k$ .

### B. General RR scheme

For general tuning, we consider the problem of locating an extremum point of the function  $C(\mathbf{p}, t): \mathbb{R}^n \times \mathbb{R}^+ \rightarrow \mathbb{R}$ , for  $\mathbf{p} = (p_1, \dots, p_n) \in \mathbb{R}^n$ , which we can measure the value of, but whose analytic form is unknown. For notational convenience, in what follows we sometimes write  $C(\mathbf{p})$  or just  $C$  instead of  $C[\mathbf{p}(t), t]$ .

The explanation presented in the previous section used  $\sin(\cdot)$  and  $\cos(\cdot)$  functions for the  $x$  and  $y$  dynamics to give circular trajectories. The actual requirement for convergence is for an independence, in the frequency domain, of the functions used to perturb different parameters, such as sines or cosines of distinct frequencies. In what follows, replacing  $\cos(\cdot)$  with  $\sin(\cdot)$  throughout, or mixing  $\sin(\cdot)$  and  $\cos(\cdot)$  terms makes no difference.

*Theorem 1.*—Consider the setup shown in Fig. 2 (for maximum seeking we replace  $k$  with  $-k$ ):

$$\dot{p}_i = \sqrt{\alpha\omega_i} \cos[\omega_i t + kC(\mathbf{p}, t)], \quad (14)$$

where  $\omega_i = \omega_0 r_i$  such that  $r_i \neq r_j \forall i \neq j$ . The trajectory of system (14) approaches the minimum of  $C(\mathbf{p}, t)$ , with its trajectory arbitrarily close to that of

$$\dot{\bar{\mathbf{p}}} = -\frac{k\alpha}{2} \nabla C, \quad \bar{\mathbf{p}}(0) = \mathbf{p}(0) \quad (15)$$

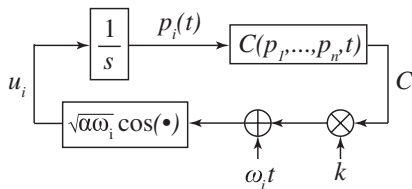


FIG. 2. Tuning of the  $i$ th component  $p_i$  of  $\mathbf{p} = (p_1, \dots, p_n) \in \mathbb{R}^n$ . The symbol  $\frac{1}{s}$  denotes the Laplace transform of an integrator, so that in the above diagram  $p_i(t) = p_i(0) + \int_0^t u_i(\tau) d\tau$ .

with the distance between the two decreasing as a function of increasing  $\omega_0$ . Namely, for any given  $T \in [0, \infty)$ , any compact set of allowable parameters  $\mathbf{p} \in K \subset \mathbb{R}^n$ , and any desired accuracy  $\delta$ , there exists  $\omega_0^*$  such that for all  $\omega_0 > \omega_0^*$ , the distance between the trajectory  $\mathbf{p}(t)$  of (14) and  $\bar{\mathbf{p}}(t)$  of (15) satisfies the bound

$$\max_{\mathbf{p}, \bar{\mathbf{p}} \in K, t \in [0, T]} \|\mathbf{p}(t) - \bar{\mathbf{p}}(t)\| < \delta. \quad (16)$$

*Proof 1.*—By expanding

$$\cos(\omega_i t + kC) = \cos(\omega_i t) \cos(kC) - \sin(\omega_i t) \sin(kC), \quad (17)$$

we rewrite the  $p_i$  ( $1 \leq i \leq n$ ) dynamics as

$$\begin{aligned} \dot{p}_i &= \sqrt{\omega_i} \cos(\omega_i t) \sqrt{\alpha} \cos(kC) \\ &\quad - \sqrt{\omega_i} \sin(\omega_i t) \sqrt{\alpha} \sin(kC), \end{aligned} \quad (18)$$

and apply corollary 1 with respect to  $\omega_0$  and  $\nu = 0.5$ . The trajectory of system (14) uniformly converges to the trajectory of

$$\begin{aligned} \dot{\bar{p}}_i &= -\frac{k\alpha}{2} \frac{\partial C(\bar{\mathbf{p}}, t)}{\partial \bar{p}_i} \{\cos^2[kC(\bar{\mathbf{p}}, t)] + \sin^2[kC(\bar{\mathbf{p}}, t)]\} \\ &= -\frac{k\alpha}{2} \frac{\partial C(\bar{\mathbf{p}}, t)}{\partial \bar{p}_i}, \end{aligned} \quad (19)$$

where we have used the fact that mismatched terms of the form  $\cos(\omega_i t) \sin(\omega_j t)$ ,  $\forall i, j$ , and terms of the form  $\cos(\omega_i t) \cos(\omega_j t)$ , and  $\sin(\omega_i t) \sin(\omega_j t)$ ,  $\forall i \neq j$  weakly, uniformly converge to zero. Combining all the  $p_i$  components we get

$$\dot{\bar{\mathbf{p}}} = -\frac{k\alpha}{2} \nabla C. \quad (20)$$

*Remark 1.*—The stability of this scheme is verified by the fact that an addition of an unmodeled, possibly destabilizing perturbation of the form  $\mathbf{f}(\mathbf{p}, t)$  to the dynamics of  $\dot{\bar{\mathbf{p}}}$  results in the averaged system:

$$\dot{\bar{\mathbf{p}}} = \mathbf{f}(\bar{\mathbf{p}}, t) - \frac{k\alpha}{2} \nabla C, \quad (21)$$

which may be made to approach the minimum of  $C$ , by choosing  $k\alpha$  large enough relative to the values of  $\|(\nabla C)^T\|$  and  $\|\mathbf{f}(\bar{\mathbf{p}}, t)\|$ . Detailed stability analysis is available in [12].

*Remark 2.*—Although it is glossed over in the averaging analysis presented above, if one looks into the details of the proof of theorem 2, in the case of a time-varying max/min location  $\mathbf{p}^*(t)$  of  $C(\mathbf{p}, t)$ , there will be terms of the form

$$\frac{1}{\sqrt{\omega}} \left| \frac{\partial C(\mathbf{p}, t)}{\partial t} \right|, \quad (22)$$

which are made to approach zero by increasing  $\omega$ . Furthermore, in the analysis of the convergence of the error  $\mathbf{p}_e(t) = \mathbf{p}(t) - \mathbf{p}^*(t)$ , there will be terms of the form

$$\frac{1}{k\alpha} \left| \frac{\partial C(\mathbf{p}, t)}{\partial t} \right|. \quad (23)$$

Together, (22) and (23) imply the intuitively obvious fact that for systems whose time variation is fast, in which the minimum towards which we are descending is quickly varying, both the value of  $\omega$  and of the product  $k\alpha$  must be larger than for the time-invariant case.

*Remark 3.*—In the case of different parameters having vastly different response characteristics and sensitivities (such as when tuning both rf and magnet settings in the same scheme), the choices of  $k$  and  $\alpha$  may be specified differently for each component  $p_i$ , as  $k_i$  and  $\alpha_i$ , without change to the above analysis.

### III. GUIDELINES FOR DIGITAL IMPLEMENTATION

#### A. Cost and constraints

The first step is to choose tunable machine parameters,  $\mathbf{p} = (p_1, \dots, p_m)$  and a cost function to be minimized,  $C = C[p_1(t), \dots, p_m(t), t]$ . Next, constraints for all parameters are chosen:

$$\mathbf{p}_{\max} = (p_{1,\max}, \dots, p_{m,\max}), \quad \mathbf{p}_{\min} = (p_{1,\min}, \dots, p_{m,\min}).$$

Implementing initial parameter settings  $\mathbf{p}(1)$ , which are chosen based on the physics model and numerical methods, allows one to calculate  $C[\mathbf{p}(1)]$ . The iterative update scheme is then

$$p_i(n+1) = p_i(n) + \Delta \sqrt{\alpha \omega_i} \cos\{\omega_i n \Delta + kC[\mathbf{p}(n)]\}, \quad (24)$$

which is based on the finite difference approximation of the derivative:

$$\frac{p_i(t+\Delta) - p_i(t)}{\Delta} \approx \frac{\partial p_i}{\partial t} = \sqrt{\alpha \omega_i} \cos\{\omega_i t + kC[\mathbf{p}(t), t]\}, \quad (25)$$

which, according theorem 1 will drive the system towards a minimum of  $C$ . The constraints are implemented by checking the updated parameters at each step and confining them to their bounds if necessary:

$$\begin{aligned} \text{IF } p_i(n+1) > p_{i,\max}, \quad & \text{THEN } p_i(n+1) = p_{i,\max}, \\ \text{IF } p_i(n+1) < p_{i,\min}, \quad & \text{THEN } p_i(n+1) = p_{i,\min}. \end{aligned}$$

#### B. Choice of $\omega$ , $\Delta$ , $k$ , and $\alpha$

It is important that  $\omega_i \gg kC$ , so that the adaptive scheme is operating on a faster time scale than and able to adapt to time variation of the cost function. Because RR depends on distinguishing between different frequency components of the cost, the  $\omega_i$  should be chosen within such bounds such that it is possible to implement a small enough  $\Delta$  obeying

$$\Delta \leq \frac{2\pi}{20 \times \max\{\omega_i\}}, \quad (26)$$

ensuring that at least 20 iterations ( $10 \times$  the Nyquist sampling rate) are required to perform one complete cosine oscillation in the iterative scheme (24). Choosing smaller values of  $\Delta$  results in smoother parameter oscillation and more iterative steps required for convergence, larger values of  $\Delta$  speed up the convergence, but may destabilize the overall scheme.

According to theorem 1, the only requirement on the choices of  $\omega_i$  is that they are big enough and distinct, but in practice, the more harmonically independent they are (such as  $\omega_i \neq 2\omega_j$  for all  $i \neq j$ ) the better. The sensitivity to frequency independence is different for every system and depends on the coupling between different components. One simple method is to choose a scaling factor,  $\omega_0$ , and  $\omega_i = \omega_0 r_i$ , where the values  $r_i$  are distinct.

The update scheme (24) is only valid as a finite difference approximation as in (25) if  $\Delta \ll 1$  and  $\omega_i \gg \sqrt{\alpha} kC$ . Taking into account (26), we choose a large value of  $\omega_0$ , relative to  $kC$ , typically at least

$$\frac{\min\{\omega_i\}}{kC[\mathbf{p}(1)]} > 20, \quad (27)$$

is a safe choice, where  $C[\mathbf{p}(1)]$  is the initial cost calculated based on initial parameter settings  $\mathbf{p}(1)$ .

The rate of convergence is proportional to the product  $k\alpha$ , increasing either  $k$  or  $\alpha$  speeds up convergence, as long as they are not too big relative to the value of  $\omega_0$ , so that the finite difference is an accurate approximation of the derivative. If, after  $\omega_0$  has been chosen, the convergence is too slow, or if a local minimum is suspected,  $k$  or  $\alpha$  may be increased, with the possible need to increase  $\omega_0$  as well. The vector  $\mathbf{p}$  is moving through the parameter space  $\mathbb{R}^m$  in ellipses with approximate major axes of magnitude  $\sqrt{\frac{\alpha}{\omega}}$ , increasing  $\alpha$  causes larger steady state parameter oscillations, which is not a problem if the adaptation is turned off following successful convergence.

The choices described above may vary from system to system based on sensitivity and initially may be an iterative process. A good approach is to first fix values of  $k$  and  $\alpha$ , define the various relationships (26) and (27), and increase  $\omega_0$  if necessary until the scheme is stable. Once convergence begins, if it is too slow, or if the cost is not sensitive enough,  $k$  and  $\alpha$  may be increased, with a possible necessity to increase  $\omega_0$  as well to maintain stability.

#### C. Digital resolution

Although the analytic form of  $C(n)$  may be unknown, at each iteration the parameters are perturbed by quantities with known bounds:

$$0 \leq |\Delta \sqrt{\alpha \omega_i} \cos\{\omega_i n \Delta + kC[\mathbf{p}(n)]\}| \leq \Delta \sqrt{\alpha \omega_{\max}}. \quad (28)$$

For a system with  $n_b$  bits of resolution, and maximum bounds  $\pm M_i$  on the parameter settings, if  $\Delta$ ,  $\alpha$ , and  $\omega_i$

are chosen such that  $\Delta\sqrt{\alpha\omega_i} \geq N \times \frac{M_i}{2^{n_b}}$ , then, as  $\cos(\cdot)$  varies between 0 and 1, it is possible for the parameter value to take  $N$  discrete steps of minimum resolution  $\frac{M_i}{2^{n_b}}$ .

#### D. Normalization of parameters

Different parameters  $p_i$  may require individual values of  $k_i$  and  $\alpha_i$ , in which case normalizing the parameters to within  $[-1, 1]$  bounds may be useful. For example, at each step  $n$ , one may compute the cost  $C(n)$  based on parameter settings  $\mathbf{p}(n)$ , then translate into the scaled parameters  $\mathbf{p}_s(n)$ :

$$p_{s,i}(n) = \frac{2[p_i(n) - C_{p,i}]}{D_{p,i}}, \quad (29)$$

where  $C_{p,i} = \frac{p_{i,\max} + p_{i,\min}}{2}$  and  $D_{p,i} = p_{i,\max} - p_{i,\min}$ , bounding each parameter within  $[-1, 1]$ . We then perform the RR update,

$$p_{s,i}(n+1) = p_{s,i}(n) + \Delta\sqrt{\alpha_i\omega_i} \cos\{\omega_i n \Delta + k_i C[\mathbf{p}(n)]\}, \quad (30)$$

force the scaled parameters to satisfy the constraints  $-1$  and  $1$ , and transform back into unscaled parameter values in order to calculate the cost for the next iteration:

$$p_i(n+1) = \frac{p_{s,i}(n+1)D_{p,i}}{2} + C_{p,i}. \quad (31)$$

#### IV. TUNING 22 QUADRUPOLE MAGNETS AND 2 BUNCHER CAVITIES

In this section we present simulation results of using the RR scheme to tune up the 22 quadrupole magnets and two buncher cavities in the Los Alamos linear accelerator  $H^+$  transport region, a simplified schematic of which is shown in Fig. 3. The simulations were done using a graphics processing unit-accelerated online beam dynamics

simulator [16,17], which is being developed to predict beam properties along the linac using real-time machine parameters. It can serve as a virtual beam experiment environment and contribute to the cost being minimized by the RR optimizer, by providing pseudo-real-time estimates of beam sizes and current information in parts of the machine where diagnostics are not available. Currently being demonstrated on the LANSCE low energy beam transport (LEBT) and drift tube linac (DTL), simulating a bunch of 32K macroparticles through the LEBT or DTL takes fractions of a second, which is 40 times faster than the simple CPU version of the code.

#### A. Magnet tuning for beam transport

In a first, simple demonstration of the technique, we perform a simulation of only the LEBT, with all initial magnet current set points set to 0A, and allowed to tune up based purely on the RR scheme as described above, in which the four costs ( $j = 1, 2, 3, 4$ ) being minimized,

$$C_j = (I_j - 0.013)^2, \quad (32)$$

were the square of the difference between initial beam current 0.013A and total current making it through various parts of the transport region, at which diagnostics are available. With reference to Fig. 3, the current is sampled at four locations,  $I_1$ , following  $Q_6$ ,  $I_2$  following  $Q_{10}$ ,  $I_3$  following  $Q_{18}$ , and  $I_4$  at the end of the transport region. The magnets ( $i = 1, \dots, 22$ ) were then updated according to

$$Q_i(n+1) = Q_i(n) + \sqrt{\alpha\omega_i}\Delta \cos[\omega_i\Delta n + kS_i(n)], \quad (33)$$

where  $S_i = C_4 + C_3 + C_2 + C_1$  for  $Q_1-Q_6$ ,  $S_i = C_4 + C_3 + C_2$  for  $Q_7-Q_{10}$ ,  $S_i = C_4 + C_3$  for  $Q_{11}-Q_{18}$ , and  $S_i = C_4$  for  $Q_{19}-Q_{22}$ , so that magnets only saw costs which they were able to influence. For the tuning parameters, we chose  $k = 250000$ , so that the amplified costs  $kS_j$  in (33) took values between 0 and 300. The  $\omega_i$  were chosen

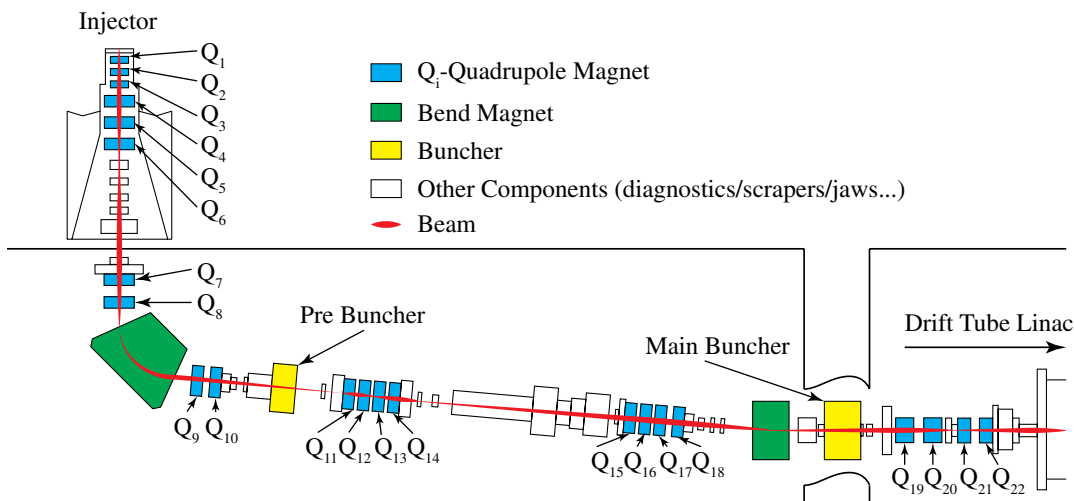


FIG. 3. Simplified schematic of the LANSCE  $H^+$  injector and transport region.

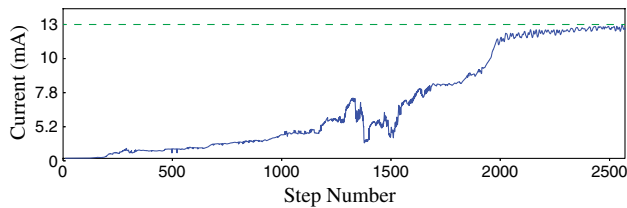


FIG. 4. The surviving current at the end of the beam transport over 2500 iteration steps is shown for an initial beam current of 13 mA.

as  $\omega_0 r_i$ , with  $\omega_0 = 1000$  and  $r_i$  uniformly distributed between 2.5 and 3.7,  $\Delta = \frac{2\pi}{20\omega_{22}}$ , and  $\alpha = 15$ . With these values,  $\frac{\omega_{\min}}{kC_{\max}} > 20$ .

Figure 4 shows the evolution of the surviving beam current at the end of the transport region during the RR tuning scheme. Figure 5 shows the evolution of the magnet current inputs. Figure 6 shows the rms beam size through various parts of the transport region at the end of RR tuning, and Fig. 7 compares the RR found magnet settings to that of the tune up in 2011.

This example demonstrates some of the strengths and limitations of the scheme, and the importance of cost function choice. Although the cost has been minimized and almost all current is making it to the end of the transport region, the beam is beginning to diverge and in this form would not be matched to the DTL following the transport region. In practice it is of course better to start with physics-model based initial parameters; this simulation was conducted starting with all magnet settings at zero

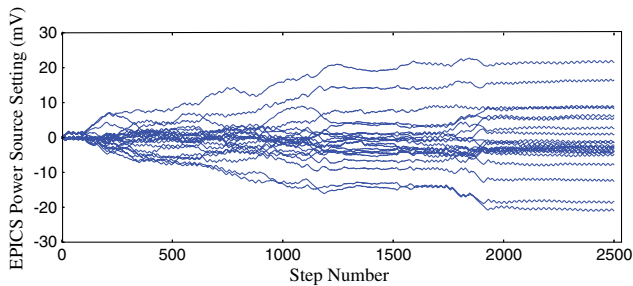


FIG. 5. Evolution of the magnet current settings to the magnets over 2500 iteration steps.

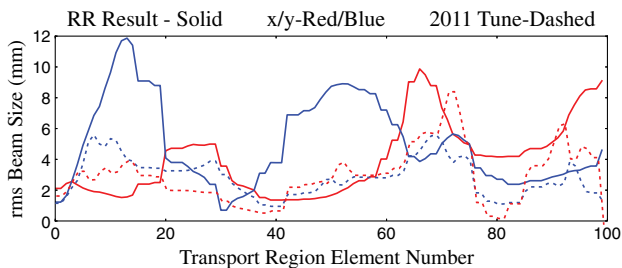


FIG. 6. Root-mean-square beam size at the end of the iterative tuning scheme.

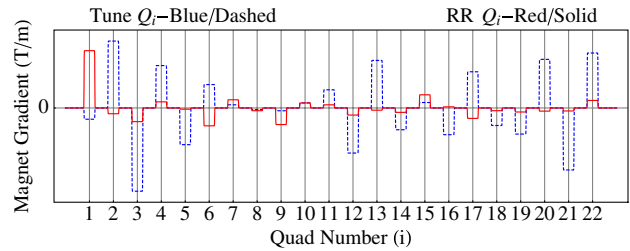


FIG. 7. Magnet settings at the end of the iterative tuning scheme compared to 2011 tune up settings.

in order to fairly demonstrate the model-independent abilities of the RR scheme. The next simulations start with the 2011 tune up for the magnet settings and use current monitors following two tanks of the DTL, in which case surviving beam corresponds with well-matched beam.

### B. Magnet and rf buncher cavity tuning

To demonstrate the use of this scheme for fine-tuning of machine settings, we used machine settings found during the 2011 tune up procedure, but with a slightly different beam and incorrectly phased buncher cavities. The magnets were initialized to the values recorded from one of the 2011 machine turn on tuning periods. We set the phase settings for the buncher and prebuncher to zero, which typically must be retuned at each turn on, by a phase scan, to take care of arbitrary phase shift.

We used only the surviving current at the end of the second tank of the drift tube linac to create our cost; our tuning procedure for the parameters was

$$Q_i(n+1) = Q_i(n) + \sqrt{\alpha_i \omega_i} \Delta \cos[\omega_i n \Delta + kC(n)], \quad (34)$$

where  $\alpha_i = \alpha_m$  for the magnets and  $\alpha_i = \alpha_b$  for the buncher phases. In both cases

$$C(n) = (I_{\text{end}} - 15 \text{ mA})^2.$$

For the tuning parameters, we chose  $k = 605000$ ,  $\alpha_m = 25$ ,  $\alpha_b = 550$ . The  $\omega_i$  were chosen as  $\omega_0 r_i$ , with  $\omega_0 = 2000$  and  $r_i$  uniformly distributed between 2.5 and 4.3,  $\Delta = \frac{2\pi}{20\omega_{24}}$ . With these values,  $\frac{\omega_{\min}}{kC_{\max}} > 35$ .

With an initial beam current of 15 mA, the typical surviving current after machine tune up is roughly 80% or 12 mA. After 2000 simultaneous iterations on these 24 parameters (22 quads, 2 buncher phases), the surviving current at the end of tank 2 was 12.25 mA. The results of the optimization procedure are shown in Figs. 8–12. From Figs. 9 and 10 we see that only minor adjustments are made to magnet settings compared to the rf phases. Figure 11 shows that the transverse beam size has further focused throughout the transport region and the transverse match to the DTL has slightly improved. Figure 12 compares surviving beam current at the end of tank 2 of the DTL before and after tuning.

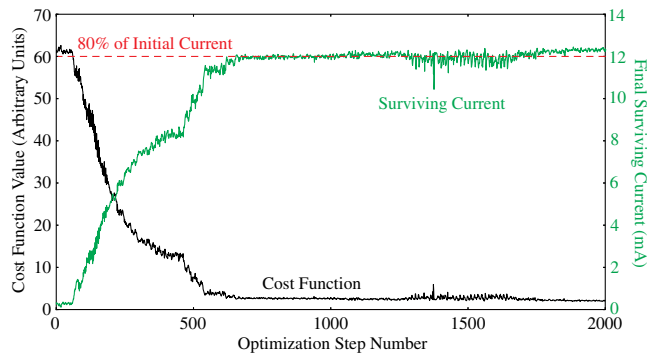


FIG. 8. The surviving current at the end of the beam transport over 2000 iteration steps is shown for an initial beam current of 15 mA.

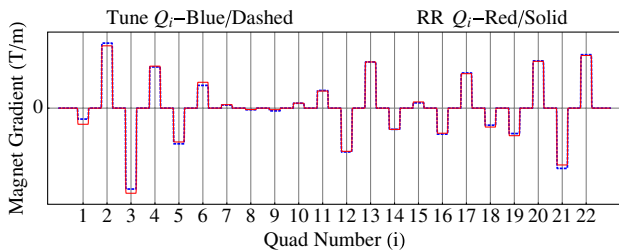


FIG. 9. New magnet settings after optimization.

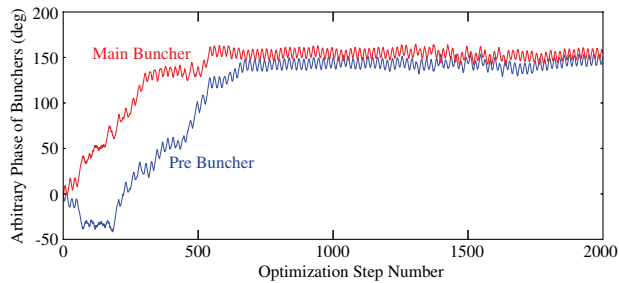


FIG. 10. Evolution of buncher cavity phase settings over 2000 iteration steps.

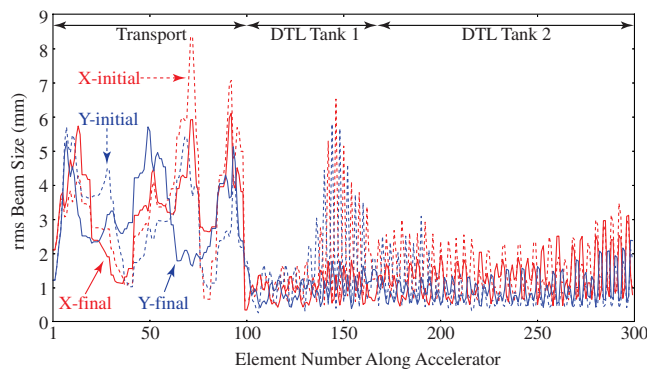


FIG. 11. Comparison of rms beam size along the accelerator for the 2011 tune-based magnet settings and arbitrary phase (dashed) with rms beam size following RR tune (solid).

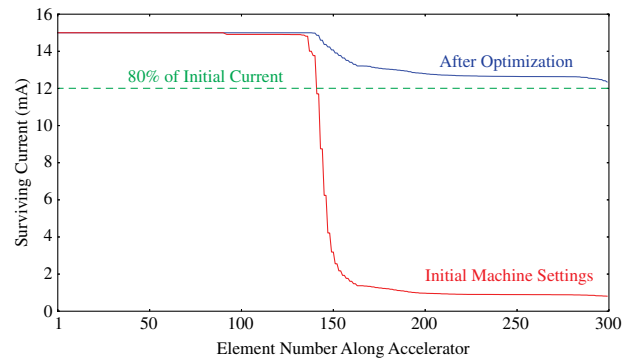


FIG. 12. Surviving beam current along the machine with 2011 tune-based magnet settings and arbitrary phase (red) and following RR tune (blue).

### C. Adaptation to time-varying phase delay and beam characteristics

In order to demonstrate the adaptive tuning abilities of the scheme, we started with matched beam settings and varied both the characteristics of the input beam and added a time-varying phase drift to each buncher cavity.

Figure 13 shows the initial and final beam properties at the entrance to the transport region, during which RR adaptive tuning maintains beam focus and matching. Figure 14 shows the phase shift of the bunchers with and without tuning. These changes took place starting at step 1000 and finished at step 19000, with beam properties staying constant before and after the interval. Also, during this beam changing process, the phase of the first buncher was made to drift by 30 deg and that of the second by 35 deg, as seen in Fig. 14.

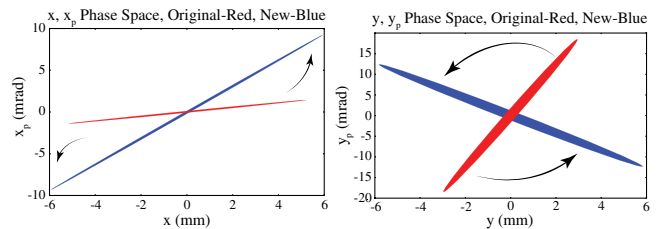


FIG. 13. The input beam was gradually changed over 18000 time steps.

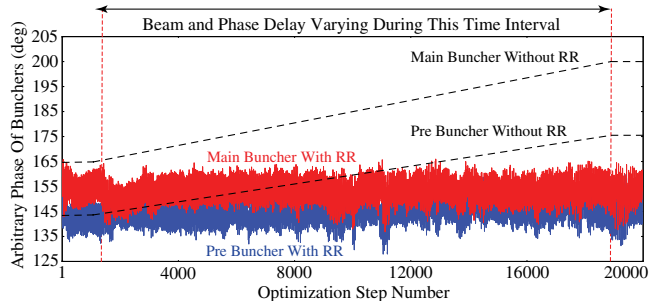


FIG. 14. Evolution of the buncher phase settings, during, and after variation of beam and phase delay parameters.



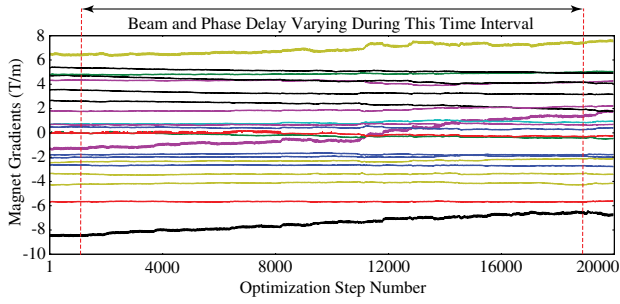


FIG. 15. Evolution of the magnet gradients before, during, and after variation of beam and phase delay parameters.

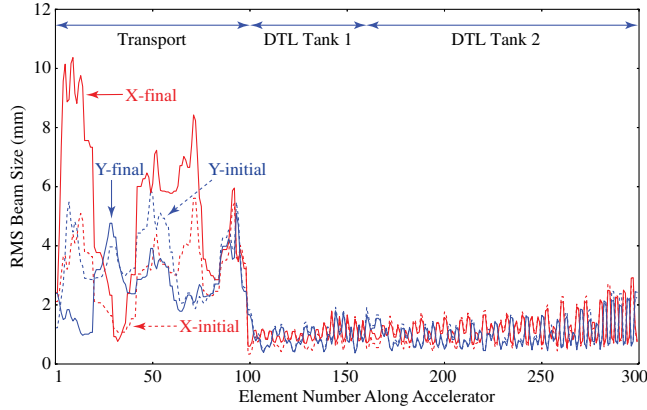


FIG. 16. Comparison of rms beam size along the accelerator for the 2011 tune-based magnet settings (dashed) and after the beam initial conditions have changed and RR tuning has refocused and matched the beam (solid).

The drift of beam characteristics and buncher phase shifts took place over 18000 time steps, which for a conservative magnet/phase update rate of 1 Hz translates into drastically changing accelerator and beam properties over the course of just 5 hours. All tuning parameters were maintained exactly the same as in the previous example.

Figure 15 shows the evolution of the magnet gradients throughout the process and Fig. 16 compares the initial and

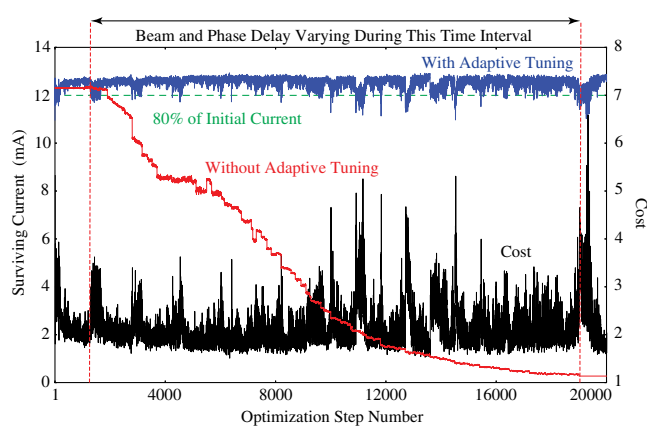


FIG. 17. Surviving beam current at the end of the second DTL tank with and without adaptive RR tuning. The cost evolution during the tuning process is also shown.

final beam profiles. In Fig. 17 we see that adaptive RR tuning is able to maintain  $\sim 12$  mA of surviving beam during the time-varying beam and phase, whereas almost all of the beam is lost without tuning.

## V. ANALYTIC BACKGROUND

We briefly recall the functional analysis result of Kurzweil and Jarnik [18], which allows one to relate the trajectories of a highly oscillatory system to those of a simplified Lie bracket averaged system.

*Theorem 2* [18].—For  $T \in [0, \infty)$ , and a compact set  $K \subset \mathbb{R}^n$ , consider a sequence ( $k \in \mathbb{N}$ ) of sets of  $n$  coupled differential equations ( $\mathbf{x} = (x_1, \dots, x_n)$ ):

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, t) + \sum_{i=1}^n \mathbf{g}_i(\mathbf{x}, t) \varphi_{i,k}(t), \quad \mathbf{x}(0) = \mathbf{x}_0, \quad (35)$$

where  $\dot{\mathbf{x}}$  denotes  $\frac{\partial \mathbf{x}}{\partial t}$  and the functions  $\mathbf{f}(\mathbf{x}, t)$ ,  $\mathbf{g}_i(\mathbf{x}, t)$ , and  $\varphi_{i,k}(t)$  are continuous and Lipschitz, and their first and second derivatives are continuous and bounded. If the functions  $\varphi_{i,k}(t)$  are continuous and their integrals satisfy

$$\Phi_{i,k}(t) = \int_0^t \varphi_{i,k}(\tau) d\tau \rightarrow 0 \quad \text{uniformly as } k \rightarrow \infty, \quad (36)$$

and there exists measurable functions  $\lambda_{i,j}(t)$  such that

$$\lim_{k \rightarrow \infty} \int_0^t \varphi_{j,k}(\tau) \Phi_{i,k}(\tau) d\tau = \int_0^t \lambda_{i,j}(\tau) d\tau, \quad \text{uniformly,} \quad (37)$$

then, for all  $t \in [0, T]$  and  $\mathbf{x} \in K$ , the sequence of solutions of (35),

$$\mathbf{x}_k(t) = \mathbf{x}_0 + \int_0^t \left( \mathbf{f}(\mathbf{x}_k, \tau) + \sum_{i=1}^n \mathbf{g}_i(\mathbf{x}_k, \tau) \varphi_{i,k}(\tau) \right) d\tau \quad (38)$$

converges uniformly with respect to  $k$ , over  $(\mathbf{x}, t) \in K \times [0, T]$  to the solution  $\mathbf{x}(t)$  satisfying

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, t) - \sum_{i,j=1}^n \lambda_{i,j}(t) [D\mathbf{g}_i(\mathbf{x}, t)] \mathbf{g}_j(\mathbf{x}, t), \quad \mathbf{x}(0) = \mathbf{x}_0. \quad (39)$$

*Corollary 1.*—For  $T \in [0, \infty)$ , and any compact set  $K \subset \mathbb{R}^n$  such that the functions  $\mathbf{f}(\mathbf{x}, t)$ ,  $\mathbf{h}_i(\mathbf{x}, t)$ ,  $\mathbf{g}_i(\mathbf{x}, t)$  satisfy the assumptions of theorem 2, for any  $\delta > 0$ , there exists  $M$  such that for all  $k > M$ , the trajectory  $\mathbf{x}(t)$  of the system

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, t) + \sum_{i=1}^n \mathbf{h}_i(\mathbf{x}, t) \hat{k}_i^\nu \cos(\hat{k}_i^\nu t) - \sum_{i=1}^n \mathbf{g}_i(\mathbf{x}, t) \hat{k}_i^\nu \sin(\hat{k}_i^\nu t), \quad (40)$$

and the trajectory  $\bar{\mathbf{x}}(t)$  of the system

$$\dot{\bar{\mathbf{x}}} = \mathbf{f}(\bar{\mathbf{x}}, t) - \frac{1}{2} \sum_{i \neq j}^n \left[ \frac{\partial \mathbf{g}_j}{\partial \bar{\mathbf{x}}} \mathbf{h}_i - \frac{\partial \mathbf{h}_i}{\partial \bar{\mathbf{x}}} \mathbf{g}_j \right], \quad \bar{\mathbf{x}}(0) = \mathbf{x}(0), \quad (41)$$

satisfy the convergent trajectories property:

$$\max_{t \in [0, T]} \|\mathbf{x}(t) - \bar{\mathbf{x}}(t)\| < \delta, \quad (42)$$

where  $k \in \mathbb{N}$ ,  $r_i \in \mathbb{R}$  such that  $r_i \neq r_j$ , and  $\hat{k}_i = r_i k$ .

*Proof 2.*—Theorem 2 is satisfied for

$$\begin{aligned} \varphi_{i,k} &= \hat{k}_i^\nu \cos(\hat{k}_i^{2\nu} t), & \Phi_{i,k}(t) &= \frac{1}{\hat{k}_i^\nu} \sin(\hat{k}_i^{2\nu} t), \\ \hat{\varphi}_{i,k} &= -\hat{k}_i^\nu \sin(\hat{k}_i^{2\nu} t), & \hat{\Phi}_{i,k}(t) &= \frac{1}{\hat{k}_i^\nu} \cos(\hat{k}_i^{2\nu} t), \end{aligned}$$

and

$$\lambda_{i,j} = \begin{cases} \frac{1}{2}: \text{mixed terms } \varphi_{i,k} \hat{\Phi}_{j,k}, \hat{\varphi}_{i,k} \Phi_{j,k} \text{ s.t. } i = j \\ 0: \text{mixed terms } \varphi_{i,k} \hat{\Phi}_{j,k}, \hat{\varphi}_{i,k} \Phi_{j,k} \text{ s.t. } i \neq j \\ 0: \text{all nonmixed terms } \varphi_{i,k} \Phi_{j,k}, \hat{\varphi}_{i,k} \hat{\Phi}_{j,k}. \end{cases}$$

## VI. CONCLUSIONS AND FUTURE WORK

Because of the global optimization ability of MOGAs and the simple, fast, abilities of RR, we think that a combination of RR and MOGA techniques can be a very powerful numerical optimization method, in which once the neighborhood of a global optimal solution has been determined by MOGA, RR may be implemented for fine-tuning, and finally in hardware to compensate for unmodeled system characteristics. In the future, following upgrades to the LANSCE digital control system and network, we plan to test this algorithm on various accelerator components.

## ACKNOWLEDGMENTS

This research was supported by Los Alamos National Laboratory.

- [1] R. Hajima, N. Taked, H. Ohashi, and M. Akiyama, *Nucl. Instrum. Methods Phys. Res., Sect. A* **318**, 822 (1992).
- [2] I. Bazarov and C. Sinclair, *Phys. Rev. ST Accel. Beams* **8**, 034202 (2005).
- [3] L. Emery, in *Proceedings of the 21st Particle Accelerator Conference, Knoxville, 2005* (IEEE, Piscataway, NJ, 2005).
- [4] M. Borland, V. Sajaev, L. Emery, and A. Xiao, in *Proceedings of the 23rd Particle Accelerator Conference, Vancouver, Canada, 2009* (IEEE, Piscataway, NJ, 2009).
- [5] L. Yang, D. Robin, F. Sannibale, C. Steier, and W. Wan, *Nucl. Instrum. Methods Phys. Res., Sect. A* **609**, 50 (2009).
- [6] A. Poklonskiy and D. Neuffer, *Int. J. Mod. Phys. A* **24**, 5 (2009).
- [7] W. Gao, L. Wang, and W. Li, *Phys. Rev. ST Accel. Beams* **14**, 094001 (2011).
- [8] R. Bartolini, M. Apollonio, and I. P. S. Martin, *Phys. Rev. ST Accel. Beams* **15**, 030701 (2012).
- [9] A. Hofler, B. Terzic, M. Kramer, A. Zvezdin, V. Morozov, Y. Roblin, F. Lin, and C. Jarvis, *Phys. Rev. ST Accel. Beams* **16**, 010101 (2013).
- [10] M. Leblanc, *The Electrification of the Railway through Alternating Currents of High Frequency*, Revue Generale de l'Electricite (1922).
- [11] W. H. Moase, C. Manzie, D. Nestic, and I. M. Y. Mareels, in *Proceedings of the 29th Chinese Control Conference*, Beijing, China, 2010.
- [12] A. Scheinker, Ph.D. thesis, University of California, San Diego, 2012 [[http://www.alexscheinker.com/Alexander\\_Scheinker\\_Thesis\\_w\\_Magnets.pdf](http://www.alexscheinker.com/Alexander_Scheinker_Thesis_w_Magnets.pdf)].
- [13] A. Scheinker, "Model Independent Beam Tuning," in *Proceedings of the 4th International Particle Accelerator Conference*, Beijing, China, 2012.
- [14] M. Borland, Report No. APS LS-287 (2000).
- [15] P. L. Kapitza, *Sov. Phys. JETP* **21**, 588592 (1951).
- [16] X. Pang, L. Rybarcyk, and S. A. Baily, in *Proceedings of the 3rd International Particle Accelerator Conference, New Orleans, Louisiana, 2012* (IEEE, Piscataway, NJ, 2012).
- [17] X. Pang and L. Rybarcyk (unpublished).
- [18] J. Kurzweil and J. Jarnik, *J. App. Math. Phys.* (1982) **38**, 241 (1987).