MODEL INDEPENDENT BEAM TUNING*

Alexander Scheinker†, LANL, Los Alamos, NM 87545, USA

Abstract

This work presents a new model independent adaptive scheme for optimization and tuning of particle accelerator components, with a simulation demonstrating the method on a low energy, space-charge dominated beam. The scheme presented here does not depend on an accurate model of the system it is stabilizing, and may even be unaware of its control input direction (such as having rotated quadrupole magnets and alignment errors) and this direction may change with time (thermal cycling/hysteresis). Stability properties are demonstrated both analytically and through a simulation in which the current settings of twenty two quadrupole magnets are simultaneously tuned through the transport section of the Los Alamos Linear Proton Accelerator. The controller is unaware of the complex nonlinear beam dynamics, with its only input being the surviving beam current readings along the transport region. Starting with all magnet settings at zero, in which case all of the beam is lost by the end of the transport, the feedback control tunes the magnets resulting in successful transport to the first drift tube linac section.

INTRODUCTION

In this work, we present a simple, model-independent tuning technique, which may be implemented in hardware to automatically fine tune multiple particle accelerator parameters simultaneously. The user first defines a cost function, C, to be minimized, which may be measured, but whose analytic form is unknown, such as the total particle loss along the length of a particle accelerator. The components p_i, of the vector $\mathbf{p} = (p_1, \ldots, p_m)$ 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 iterative tuning law is then given by:

$$p_i(n+1) = p_i(n) + \Delta\sqrt{\alpha\omega_i}\cos\left(\omega_i \Delta n + kC(\mathbf{p}(n))\right). \tag{1}$$

Initial settings $\mathbf{p}(1)$, may be chosen as usual, based on a physics model. 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). Constraints are easy to implement by defining upper and lower bounds for each parameter and implementing:

$$\textbf{IF } p_i(n+1) > p_{i,\max}, \quad \textbf{THEN } p_i(n+1) = p_{i,\max},$$
$$\textbf{IF } p_i(n+1) < p_{i,\min}, \quad \textbf{THEN } p_i(n+1) = p_{i,\min}.$$

Because this method is model independent, it may be useful in helping mitigate un-modeled disturbances and component imperfections.

TUNING METHOD

Physical Motivation

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:

$$\frac{\partial x}{\partial t} = \sqrt{\alpha\omega}\cos\left(\omega t + kC(x,y)\right) \tag{2}$$
$$\frac{\partial y}{\partial t} = \sqrt{\alpha\omega}\sin\left(\omega t + kC(x,y)\right). \tag{3}$$

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 it's value is available for measurement at different locations (x, y).

The velocity vector

$$\mathbf{v} = \left(\frac{\partial x}{\partial t}, \frac{\partial y}{\partial t}\right) = \sqrt{\alpha\omega}\left[\cos\left(\theta(t)\right), \sin\left(\theta(t)\right)\right], \tag{4}$$

where

$$\theta(t) = \omega t + kC(x(t), y(t)), \tag{5}$$

has constant magnitude

$$\|\mathbf{v}\| = \sqrt{\alpha\omega}, \tag{6}$$

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 ω, k, and $C(x(t), y(t))$:

$$\frac{\partial\theta}{\partial t} = \omega + k\frac{\partial C}{\partial t} = \omega + k\left(\frac{\partial C}{\partial x}\frac{\partial x}{\partial t} + \frac{\partial C}{\partial y}\frac{\partial y}{\partial t}\right). \tag{7}$$

Therefore, when the trajectory is heading in the correct direction, towards a decreasing value of $C(x(t), y(t))$, the term $k\frac{\partial C}{\partial t}$ is negative, and so the overall turning rate $\frac{\partial\theta}{\partial t}$, (7) is decreased. When, on the other hand, the trajectory is heading in the wrong direction, towards an increasing value of $C(x(t), y(t))$, the term $k\frac{\partial C}{\partial t}$ 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.

* Funding supported by Los Alamos National Laboratory
† ascheink@lanl.gov

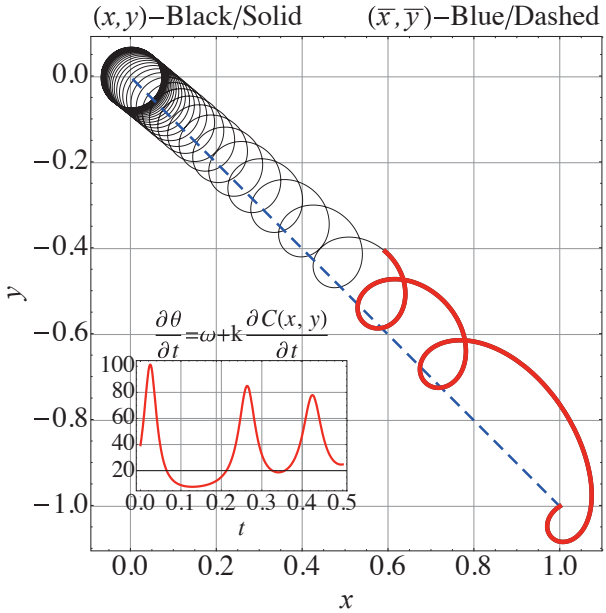

Figure 1: The subfigure shows the rotation rate, $\frac{\partial\theta}{\partial t} = \omega + \frac{\partial C(x,y)}{\partial t}$ in red, for the part of the trajectory that is in bold red, the first 0.5 seconds of simulation. The rotation of the parameters' velocity vector $\mathbf{v}(t)$ slows down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

The ability of this direction-dependent turning rate scheme is apparent in the simulation of system (2), (3), in Figure 1. The system, starting at initial location $x(0) = 1$, $y(0) = -1$, is simulated for 5 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 (2), (3) dynamics with those of a system performing gradient descent:

$$\frac{\partial \bar{x}}{\partial t} \approx -\frac{k\alpha}{2}\frac{\partial C(\bar{x},\bar{y})}{\partial \bar{x}} = -k\alpha\bar{x} \quad (8)$$

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

whose behavior our system mimics on average, with the difference

$$\max_{t\in[0,T]} \|(x(t),y(t)) - (\bar{x}(t),\bar{y}(t))\| \quad (10)$$

made arbitrarily small for any value of T, by choosing arbitrarily large values of ω. The derivation of this relationship and of the rate of the gradient descent are based on Theorem 1 and discussed in more detail in [1]. Towards the end of the simulation, when the system has approached the origin, so that $C(x,y) \approx 0$, the dynamics of (2), (3) are approximately

$$\frac{\partial x}{\partial t} \approx \sqrt{\alpha\omega}\cos(\omega t) \Longrightarrow x(t) \approx \sqrt{\frac{\alpha}{\omega}}\sin(\omega t)$$

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

which is a circle of radius $\sqrt{\frac{\alpha}{\omega}}$, which can be made arbitrarily small for a fixed value of α by choosing arbitrarily large values of ω. Convergence towards a maximum, rather than a minimum is achieved by replacing k with $-k$.

For convergence to take place, the perturbing functions of the different parameters must be independent, in the frequency domain. For a given system, with a cost function $C(p_1,\ldots,p_n,t)$, if we perturb the parameters according to (replacing $\cos(\cdot)$ with $\sin(\cdot)$ below makes no difference):

$$\dot{p}_i = \sqrt{\alpha\omega_i}\cos(\omega_i t + kC(p_1,\ldots,p_n,t)) \quad (11)$$

such that the ω_i are distinct, we get the average behavior

$$\dot{\bar{p}}_i = -\frac{k\alpha}{2}\frac{\partial C}{\partial \bar{p}_i}. \quad (12)$$

In practice, the degree to which the ω_i are distinct is important, while $\omega_1 = 50, \omega_2 = 50.1$ is usually a bad choice, $\omega_1 = 50, \omega_2 = 77$ is much better, the sensitivity is different for every system/cost function that the method is applied to and depends on coupling between different components.

SIMULTANEOUSLY TUNING 22 QUADRUPOLE MAGNETS

We perform a simulation of the low energy H^+ beam transport section of the Los Alamos LANSCE accelerator, with all initial magnet current settings set to $0A$, and allowed to tune up based purely on the scheme as described above, in which the four costs (j=1,2,3,4) being minimized:

$$C_j = (I_j - 0.013)^2, \quad (13)$$

are 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 Figure 2, I_1 is the surviving beam current preceding Q_7, I_2 is I_1 added to current measurement following Q_{14}, I_3 is I_2 added to a measurement in front of Q_{15} and I_4 is measured following Q_{22}. The magnets were then updated according to:

$$Q_i(n+1) = Q_i(n) + \sqrt{\alpha\omega_i}\Delta\cos(\omega_i\Delta n + kC_j(n)), \quad (14)$$

where different magnets only saw costs which they were able to influence. Figure 4 shows the evolution of the magnet settings, Figure 3 shows the surviving beam current at the end of the transform throughout the tuning process, and Figure 5 shows the RMS beam sizes at the end of the tuning procedure.

ISBN 978-3-95450-122-9

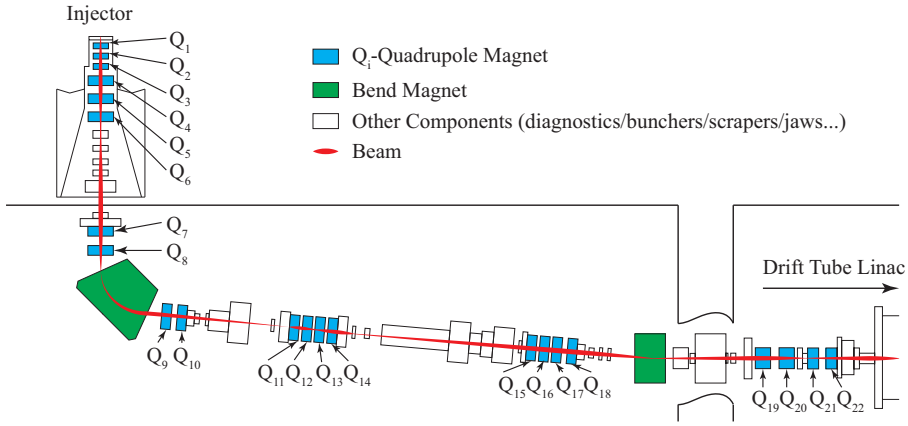

Figure 2: Simplified schematic of the LANSCE H^+ injector and transport region.

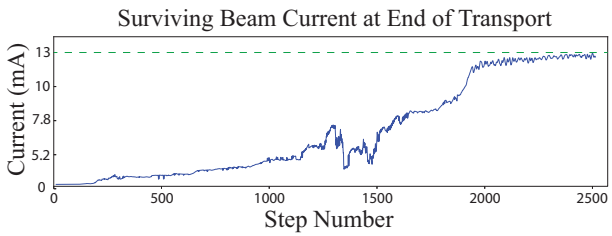

Figure 3: The surviving current at the end of the beam transport over 2500 iteration steps is shown for an initial beam current of 13mA.

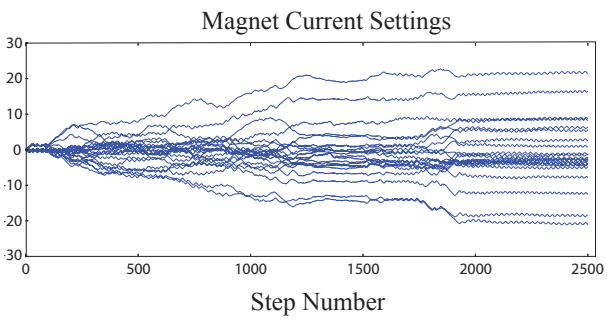

Figure 4: Evolution of the current settings to the magnets over 2500 iteration steps.

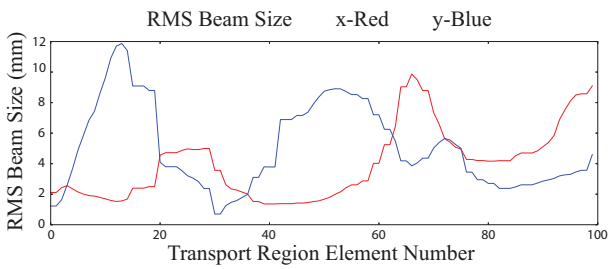

Figure 5: RMS beam size at the end of the iterative tuning scheme.

ANALYTIC BACKGROUND

Our averaging analysis is based on a functional analysis result of Kurzweil and Jarnik [3], which allows one to relate the trajectories of a highly oscillatory system to those of a simplified Lie bracket averaged system.

Theorem 1 *[3] For $T \in [0, \infty)$, and any compact set $K \subset \mathbf{R}^n$ such that the functions $\mathbf{f}(\mathbf{x}, t)$, $\mathbf{h}_i(\mathbf{x}, t)$, $\mathbf{g}_i(\mathbf{x}, t)$ are continuous and continuously differentiable, for any $\nu, \delta > 0$, there exists M such that for all $k > M$, the trajectory $\mathbf{x}(t)$ of the system*

$$\begin{aligned}\dot{\mathbf{x}} &= \mathbf{f}(\mathbf{x}, t) + \sum_{i=1}^{n} \mathbf{h}_i(\mathbf{x}, t)(kk_i)^{\nu} \cos\left((kk_i)^{2\nu} t\right) \\ &\quad - \sum_{i=1}^{n} \mathbf{g}_i(\mathbf{x}, t)(kk_i)^{\nu} \sin\left((kk_i)^{2\nu} t\right), \end{aligned} \tag{15}$$

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

$$\dot{\bar{\mathbf{x}}} = \mathbf{f}(\bar{\mathbf{x}}, t) - \frac{1}{2} \sum_{i \neq j}^{n} \left[\mathbf{h}_i(\bar{\mathbf{x}}, t), \mathbf{g}_j(\bar{\mathbf{x}}, t)\right], \quad \bar{\mathbf{x}}(0) = \mathbf{x}(0), \tag{16}$$

satisfy the convergent trajectories property:

$$\max_{t \in [0,T]} \|\mathbf{x}(t) - \bar{\mathbf{x}}(t)\| < \delta, \tag{17}$$

where $k \in \mathbf{N}$, $k_i \in \mathbf{R}$ such that $\hat{k}_i \neq \hat{k}_j$, and

$$[\mathbf{h}_i(\bar{\mathbf{x}}, t), \mathbf{g}_j(\bar{\mathbf{x}}, t)] = \frac{\partial \mathbf{g}_j}{\partial \bar{\mathbf{x}}} \mathbf{h}_i - \frac{\partial \mathbf{h}_i}{\partial \bar{\mathbf{x}}} \mathbf{g}_j. \tag{18}$$

ACKNOWLEDGEMENTS

The author is grateful to Xiaoying Pang and Larry Rybarcyk for the use of their simulation software [2], and Miroslav Krstic for discussion of the adaptive algorithm.

REFERENCES

[1] A. Scheinker, Ph.D. thesis, University of California, San Diego, November, 2012. Available online at: http://www.alexscheinker.com/research

[2] X. Pang, L. Rybarcyk, and S. A. Baily, "High-performance beam simulator for the LANSCE linac," *Proc. IPAC*, 2012.

[3] J. Kurzweil and J. Jarnik, "Limit processes in ordinary differential equations," *Journal of Applied Mathematics and Physics* vol. 38, March 1987, pp. 241-256.