FAST BEAM TUNING FOR ACCELERATOR DRIVEN SYSTEMS

S. Bhattacharyya, R. K Yedavalli, The Ohio State University, Columbus, OH 43201, USA
A. Mukherjee, Fermilab, Batavia, IL 60510, USA

Abstract

A future application of linear accelerators (linacs) in Nuclear Energy is Accelerator Driven Subcritical Systems (ADSS, or Accelerator Driven Systems, ADS). ADS is a revolutionary concept where a proton linac produces energy from transmutation of conventional radioactive waste[1]. The major challenge in realizing this concept is combining high efficiency and high reliability in a proton accelerator. High reliability demands for automated controls and fast online repairs in the accelerator. In this paper we talk about how fast beam tuning can be achieved by automated controls. We develop beam model using State Space Method so that traditional control techniques can be applied. Further we discuss the use of χ^2 Minimization Technique for actual beam control.

CONTROLS FOR ADS

Importance

Any kind of failure, or repair, or magnet drifts in an accelerator cause beam interruptions, which is one of the major concern in Accelerator Driven Systems. The high intensity 1 MW beam in ADS strikes a small area~1cm^2 on the Spallation Target (which produces external neutrons for fission). Frequent beam interruptions leads to alternative cooling and heating of the spot, leading to thermal stress on the spallation target. This damages the Spallation target, and eventually cause harm to the Fission Target as well. Replacing the targets is extremely expensive-and requires shutting down the plant for a considerable period. This means, ADS can take a bunch of tiny interruptions (<2 seconds) or a very few medium or large interruptions. A large number of medium beam interruptions of greater than 2 seconds give rise to thermal shock on the targets. Therefore, we need a very fast controller that can automatically bring the beam back to its nominal when it suffers a deviation due to any abnormality in the system within this time scale. In this paper, we discuss the methodologies that can be used to model the beam so that it can be used eventually by a traditional controller to implement this objective.

Present Techniques

In Tevatron, Fermilab's main accelerator, beam tuning was done manually. Though the total emittance of the beam is constant (almost), and with variation of the magnet current, the beam size in X and Y varies. Beam emittance can be measured by observing:

- Lateral profile
- Longitudinal profile
- Beam intensity

A Laser scanner, as shown in Figure 1, and a pick up coil are used for this purpose. The data from the laser scan is then used to tune the magnets manually which help to bend the particles to the beam line. This can take several hours to days.

Further accelerators are also provided the beam position monitors or BPMs that gives the position of the centroid of the beam. Most accelerators have a much larger number of BPMs and only a few laser scanners.

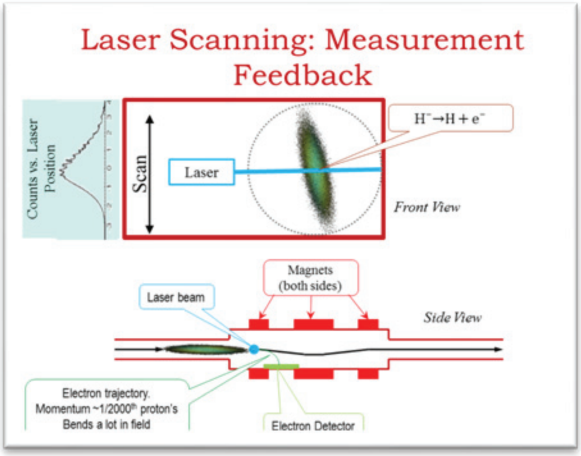

Figure 1: Laser Scanner for measuring beam profile.

Some work on automated feedback control system has been done in NUMI for maintaining desired trajectory. Controls systems for maintaining beam trajectory using BPMs is not uncommon. Our objective is to design a control technique maintain the beam size and make sure, at no point the beam blows up and touches the cavity walls.

BEAM MODELING

State Space Method of Beam Representation

Before applying controls technique, a beam model is necessary. We propose to model the beam in a form widely prevalent in the controls field: State Space Method. We define state of dynamic system as the smallest set of linearly independent variables, called the "state variables" such that the knowledge of these variable at $t = t_0$, together with the input at $t \geq t_0$ completely determines the behaviour of the system for any time $t \geq t_0$. State space method uses matrix representation, and in time domain can be expressed in the following form

$\dot{x}(t) = Ax(t) + Bu(t)$ in continuous time domain

$x(k+1) = Ax(k) + Bu(k)$ in discrete domain where, x is called the state vector, u is the input or control vector, A is called the plant matrix, and B is the input or control matrix.

To model the beam, we at first simulated a small section of a high intensity proton linac using TraceWin, a software for accelerator simulation. The segment consists of 5 quadrapoles along with two gaps, and drift regions between the magnets. We intend to represent the beam in form of equation (2),

$$\vec{v}(k+1) = \boldsymbol{A}\vec{v}(k) + \boldsymbol{B}\vec{u}(k)$$

where $\boldsymbol{A}$ is the plant matrix and $\boldsymbol{B}$ is the control matrix. Note we use $\vec{v}$ rather than the more conventional $\vec{x}$ to avoid confusion with the x coordinate.Here k denotes the position index corresponding to z. For controls, the dynamics of the beam is too fast with time, so we are first interested in the variation with position and hence use the discrete state space form [2].

In the first test, we turn off all the magnet currents. Therefore, $\vec{u} = 0$. We take $\vec{v}$ as

$$\vec{v}(k) = \begin{pmatrix} B_x(k) \\ B_y(k) \end{pmatrix}$$

where B_x, B_y are the beam size in the x and y directions, and k is the index for position along the beam direction, i.e. z. We then have Excel fit the 2×2 matrix $\boldsymbol{A}$ and the initial value, $\vec{x}(0)$. The resulting fit is quite poor, as seen in Figure 2.

The problem can be understood as follows. With $\vec{u} = 0$

$$\vec{v}(k) = \boldsymbol{A}\vec{v}(k-1) = \boldsymbol{A}^2\vec{v}(k-2) = \cdots = \boldsymbol{A}^k\vec{v}(0)$$

With magnets off, physically there is no source of coupling between B_x and B_y so $\boldsymbol{A}$ should be diagonal. Then

$$B_x(k) = (A_{11})^k B_x(0)$$
$$B_y(k) = (A_{22})^k B_y(0)$$

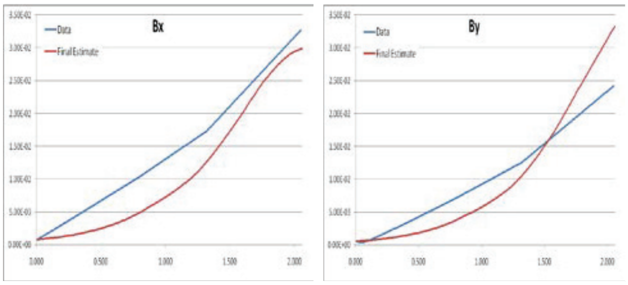

Figure 2: 2D model with zero current.

i.e., the growth is exponential. Although the fit is somewhat improved by allowing $\boldsymbol{A}$ to be non-diagonal.

4D Linear Model, no Magnet

To allow a wider range of functions, in particular linear, we expanded $\vec{v}$ to a 4-vector

$$\vec{v}(k) = \begin{pmatrix} B_x(k) \\ S_x(k) \\ B_y(k) \\ S_y(k) \end{pmatrix}$$

where S_x, S_y are the rate of change of B_x, B_y. This allows linear result, for instance using

$$\boldsymbol{A} = \boldsymbol{A_0} = \begin{pmatrix} 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

yields

$$S_x(k) = S_x(k-1) = \cdots = S_x(0)$$
$$B_x(k) = B_x(k-1) + S_x(k) = \cdots B_x(0) + kS_x(0)$$

and similarly for y.

Keeping $\boldsymbol{A}$ fixed at $\boldsymbol{A}_0$ and allowing Excel to fit only $\vec{v}(0)$ gives a line fit as seen below. The $\mathrm{B_x}$ response is showed in Figure 3, and that of $\mathrm{B_y}$ was similar. Because of the non-linear behaviour at the beginning (where the beam is converging), and the kink near 1.3m (due to an acceleration cavity), the fit is not great. Allowing Excel to fit $\boldsymbol{A}$ (now 4×4), as well as $\vec{v}(0)$, we got a better fit.

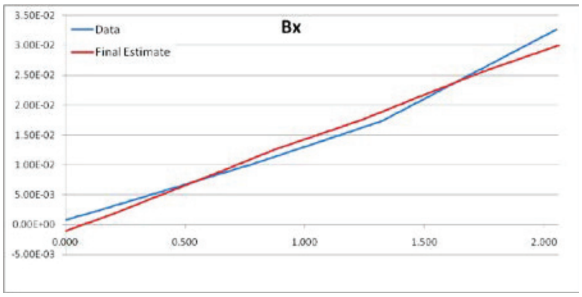

Figure 3: 4D linear model, with zero current.

4D Linear Model, with Magnets

Next we work on the case with magnets at nominal settings. To temporarily bypass how to set $\vec{u}$ and $\boldsymbol{B}$, we start of assuming we can perfectly model regions with magnets and accelerating gaps. We implement this by assigning

$$\vec{v}(k) = \vec{\tilde{v}}(k)$$

everywhere except drift regions, where $\vec{\tilde{v}}(k)$ is the data.

Note if the model is correct, $\boldsymbol{A}$ must be independent of magnet (control) settings. Using the matrix fit to the magnets-off data to nominal magnet settings gives the following; using $\boldsymbol{A} = \boldsymbol{A}_0$ gives essentially identical results. We show the $\mathrm{B_y}$ results in Figure 4, to demonstrate the failure of the model in the region of the focal point. Refitting in excel though gave some improvement, but that same A failed to work in the zero current situation.

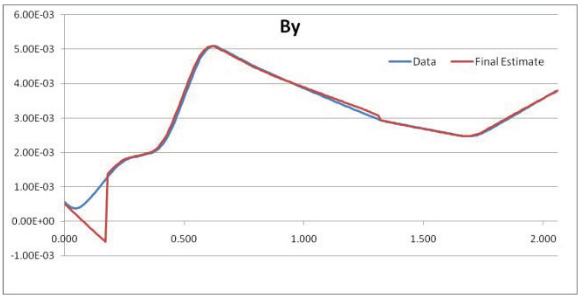

Figure 4: 4D Linear Model with magnets.

Non-linear Model

We found that a linear model works well for magnet-off data, but poorly for magnet-on data. Furthermore, the optimal $\boldsymbol{A}$ matrix is different for the two cases. We therefore turn to non-linear modeling.

In this model, we are going to do an addition in quadrature.

$$a \oplus b \overset{\text{def}}{=} \sqrt{a^2 + b^2}$$

We start with a linear model like before, except label the components L instead of B to distinguish the linear aspect

$$\vec{v}(k) = \boldsymbol{A}\vec{v}(k-1)$$

To get final beam size estimates, we next add a constant to it:

$$B_x(k) = L_x(k) \oplus s$$

We do this because we note that, except at a focal point, the beam between magnets and gaps shows linear growth or shrinkage. At a focal point the linear term would go to zero and then flip sign , that is the size would become negative. We want to turn this into something which is always positive, and has a minimum value. This is one fairly simple way to do that.

Using $\boldsymbol{A} = \boldsymbol{A}_0$ and fitting only s gave good fit to both magnet-off and nominal data. In Figure 4 we show the $\mathrm{B_y}$ results again, and how the focal point response is now fixed.

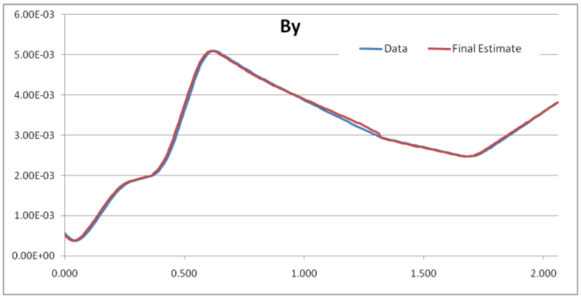

Figure 5: Non Linear Model with magnets.

So far we were only modelling the drift regions in our linac section. We now try to introduce the effect of the magnet and model the beam throughout the linac. In this case we find that our B is varying with k, and can be expressed as but we define B as a 4x1 vector, having elements:

$$B(k) = \begin{pmatrix} 0 \\ Sx \\ 0 \\ Sy \end{pmatrix}$$

So we only apply correction to the slopes. The B computed is a function of k, that it is different for each k. u is taken as a scalar, that is the magnet current at that position k. So,

$$u(k) = i \; for \; k \neq drift \;\; and \; u(k) = 0 \; for \; k = drift$$

CHI –SQUARE MINIMIZATION

So far we only talked about beam modelling, but it is useful to mention some of the control algorithms we applied to bring the beam back to normal after it blows up due to magnet drifts. We consider in this example multiple magnets suffered slow drifts over the time. In this case we minimize

$$\chi^2 = \sum_j \frac{\left(I_j - \tilde{I}_j\right)^2}{\sigma_j^2} + \sum_k \frac{\left(B_k - \tilde{B}_k\right)^2}{\sigma_k^2}$$

I_j is the measured current for magnet $j, \tilde{I}_j$ is the nominal current for magnet j, σ_j is the expected uncertainty in I_j, B_k is a measured beam property (size), $\tilde{B}_k$ is the nominal value of B_k, σ_k is the expected uncertainty in B_k.

We get the following result in Figure 6 where the blue is the nominal, red is the beam due to magnet drift and green is the corrected beam by chi-square method.

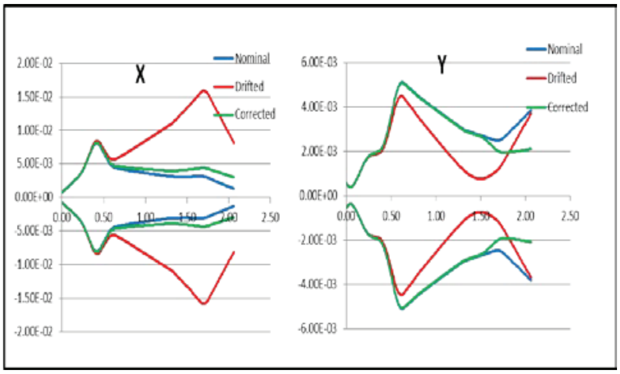

Figure 5: Beam Control using Chi Square Minimization.

REFERENCES

[1] Abderrahim *et al.*, "Accelerator and Target Technology for Accelerator Driven Transmutation and Energy Production", (2010)

[2] Roesser, R.B. (1975), "A discrete state space model for linear image processing", IEEE. Transactions on Automatic Control, Vol. AC-20, pp. 1-10.