

SLAC-PUB-7398

February 1997

## Feedback: Theory and Accelerator Applications\*

T. Himel

Stanford Linear Accelerator Center  
Stanford University, Stanford, CA 94309

### Abstract

The use of feedback to stabilize the beam and improve the performance of accelerators is becoming more common. The methods used to design the feedback algorithms are introduced and some practical implementation details are described. The design of a PID loop using classical control techniques is covered as is the design of an optimal controller using modern control theory. Some adaptive control techniques are also briefly described. Examples are given of multiple-input-multiple-output loops and of how to handle systems of many interacting feedback loops.

Submitted to *Annual Review of Nuclear and Particle Science*

---

\*Work supported by Department of Energy contract DE-AC03-76SF00515.

# Contents

<b>1</b>	<b>INTRODUCTION</b>	<b>3</b>
<b>2</b>	<b>BASIC IDEAS AND DEFINITIONS</b>	<b>6</b>
2.1	<i>A Simple Example: the Simple Harmonic Oscillator . . .</i>	6
2.2	<i>Time Invariant Linear System . . . . .</i>	6
2.3	<i>State-Space Notation . . . . .</i>	8
2.4	<i>The Transfer Function . . . . .</i>	10
2.5	<i>Feedback and Feed-forward . . . . .</i>	12
2.6	<i>SISO and MIMO loops . . . . .</i>	15
2.7	<i>Analog and Digital Feedback . . . . .</i>	15
2.8	<i>Continuous and Discrete Systems . . . . .</i>	16
2.9	<i>Disturbance Rejection vs. Tracking Reference Changes .</i>	17
2.10	<i>Modeling the Noise . . . . .</i>	18
<b>3</b>	<b>FEEDBACK DESIGN FORMALISMS</b>	<b>20</b>
3.1	<i>Classical Control Theory . . . . .</i>	21
3.1.1	PROPORTIONAL CONTROL . . . . .	21
3.1.2	INTEGRAL CONTROL . . . . .	23
3.1.3	DERIVATIVE CONTROL . . . . .	23
3.1.4	PID CONTROL . . . . .	24
3.2	<i>Modern Control Theory . . . . .</i>	27
3.3	<i>Adaptive Filters and Control . . . . .</i>	29
<b>4</b>	<b>TRADEOFFS</b>	<b>31</b>
4.1	<i>Robustness vs. Performance . . . . .</i>	31
4.2	<i>Performance vs. Dependence on Exact Knowledge of the Noise Spectrum . . . . .</i>	32
4.3	<i>Actuator Strength vs. Speed and Cost . . . . .</i>	32
<b>5</b>	<b>MULTIPLE LOOP INTERACTIONS</b>	<b>33</b>
5.1	<i>Example: Global Steering Feedback at APS . . . . .</i>	34
5.2	<i>Example: Adaptive Cascade at SLAC . . . . .</i>	35
<b>6</b>	<b>SOME POTENTIAL PROBLEMS AND COUNTER- MEASURES</b>	<b>38</b>
6.1	<i>Aliasing . . . . .</i>	38
6.2	<i>Quantization Noise . . . . .</i>	39
6.3	<i>Actuator Saturation . . . . .</i>	39

## 1 INTRODUCTION

The use of beam based feedback in accelerators has been gradually growing as the field advances. It has gone from an afterthought, added on to slightly improve the stability or performance of an accelerator, to a system which is incorporated into the design from the beginning. Accelerators are being designed and constructed that would not be at all practical without their feedback systems. Hence a knowledge of feedback systems is becoming as essential in the design of an accelerator as a knowledge of optics, particle tracking, magnet design, vacuum systems, and RF systems.

Accelerator feedback systems fall into many broad categories and it is not possible to cover all of them in this paper. Probably the most numerous ones are in the magnet power supplies. These loops sense the current heading to the magnet and control the voltage of a supply to keep it at the desired value. While the accelerator physicist must know what tolerances can be achieved (and at what expense) with these regulator loops, their design is well described in books [1] and can typically be left to a good electronics engineer and hence will not be covered in this paper. Another common application of feedback in accelerators is in the RF systems of storage rings. There may be up to five interrelated feedback loops designed to maintain the fields in the RF cavities at desired values. In a high current accelerator, beam loading has a major effect on these fields and the situation gets quite complicated. Fortunately there are already excellent review papers [2] on RF feedback systems, so we can safely omit them from this article.

This article will concentrate on feedback systems which directly measure and control some property of the beam. These are generically referred to as beam based feedback loops. Examples abound:

- A steering loop uses beam position monitors (BPMs) to measure the position and angle of the beam in a transport line. It then uses steering magnets to correct the orbit [3].
- In a synchrotron light ring, the position and angles of the X-ray beams are measured and then corrected using steering magnets in the ring [4, 5, 6, 7, 8, 9].

- At the end of a linear accelerator, the beam energy is measured with BPMs in a spectrometer. That energy is then corrected by varying the phases or amplitudes of the RF accelerating system [10].
- The electron beam intensity is measured downstream from a photocathode gun. The laser intensity is varied to obtain the desired intensity.
- The transverse tune of a storage ring is measured and then corrected by varying the quadrupole string currents [11].
- The phase of each bunch in a storage ring is measured. Small energy kicks are then supplied to damp out oscillations in the phase [12, 13].
- The position of each bunch in a storage ring is measured. Small transverse kicks are then supplied to damp out oscillations [14, 15, 16].

While feedback has been applied to many accelerator problems, only a few basic terms and techniques need to be learned to understand how they all work and, in fact, to design your own system. In this paper, we will cover these basic terms and techniques. A simple harmonic oscillator will be used as an example throughout the paper. As it turns out, many real-life problems are in fact simple harmonic oscillators, so this example can be directly applied to many accelerator problems. We will also illustrate ideas with examples from various real feedback loops implemented in accelerators around the world. In addition to the basic ideas, explanations will be given about various complications that occur and how they have been solved.

The rest of this paper is structured as follows: The next section defines the basic terms used in the field. Following that is an exposition of the various types of feedback methods: classical, modern, and adaptive. The advantages and disadvantages of each approach are explained. Then examples are given of some of the basic tradeoffs which must be made when designing a real system. The succeeding section is devoted to problems which occur in accelerators with multiple interacting loops and the interesting and innovative solutions found at several laboratories. The penultimate section points out a few technical problems which can occur and gives ideas on how to deal with them. The paper concludes with a summary.

Before continuing on, it is interesting to ask why feedback systems are needed at all. There are two main reasons. Some systems are unstable without feedback. A small perturbation would grow exponentially until the beam is lost. For example, the wake fields in the high current B-factory storage rings are expected to cause unstable coupled bunch motions [17] that can only be stabilized with feedback. The second reason is to compensate for uncontrolled drifts and oscillations of accelerator components. Temperature variations may move magnets which in turn move the beam. Vibrations due to water pumps or ground motion can shake magnets and move the beam. Magnet power supplies or RF phases can drift and disturb the beam. Operators may adjust components which perturb the beam. The list is endless. While it is best to eliminate such problems at their source, in practice this may be impossible, impractical, or too expensive, so a feedback loop is used instead. In such cases, the beam without feedback is not technically unstable; that is, perturbations do not grow exponentially. Rather, the motion is a random walk or gradual drift due to all the things which are varying.

Why is it that beam based feedback systems are becoming more common? Two reasons come to mind. The accelerator field is maturing. Third and fourth generation accelerators are being designed and built. More tools are needed to meet the higher requirements of each generation. Feedback is one of these tools. The second reason is that advancing technology has made it far easier (and cheaper) to design and implement feedback loops. There are tools which will run on your PC or workstation that allow you to draw a block diagram of your system. MATRIXx by Integrated Systems Inc., Matlab with Simulink by the MathWorks Inc., CC by Systems Technology Inc., and Ctrl-C with Model-C by Systems Control Technology Inc. are four such products. These products can then design an optimum feedback loop for your system, simulate it, and plot its transfer function. It is even possible to get the hardware for a prototyping system, have the design tool load the software into the digital signal processor (DSP) of that system, and test a feedback loop on your real accelerator. These tools have greatly reduced the time and expertise needed to implement a feedback loop. So, while the control theory used for feedback has been around for quite a few years, the advanced computerized design tools have brought its use out of the domain of industries with very large R+D budgets, such as the aerospace industry, into the domain where it is reasonable to design individualized special purpose loops.

## 2 BASIC IDEAS AND DEFINITIONS

### 2.1 *A Simple Example: the Simple Harmonic Oscillator*

Throughout this paper, we will use the simple harmonic oscillator (SHO) as an example of a system where feedback can be applied. Examples of SHOs in accelerators are the transverse and longitudinal motions of the beam in a storage ring. For our example here, we will use the most familiar oscillator of all, a mass of mass  $m$  hanging on a spring with spring constant  $k$  with a damping force  $\gamma$  proportional to the velocity of the mass. The differential equation for this system is

$$m\ddot{z} + \gamma\dot{z} + kz = F(t) \quad (1)$$

where  $F(t)$  is the force on the mass as a function of time. The force is applied in two ways as shown in Figure 1.

- By displacing the top of the spring a distance  $d$ . This is something under our control and will later be used as the actuator in a feedback loop.
- By an uncontrolled disturbing force,  $f$ , perhaps air currents, as represented by the fan in the figure.

The differential equation now becomes

$$\ddot{z} + \frac{\gamma}{m}\dot{z} + \frac{k}{m}z = \frac{k}{m}d + \frac{f}{m}. \quad (2)$$

Now define the resonant frequency and quality factor of the oscillator as follows:

$$\omega_0 = \sqrt{\frac{k}{m}} \quad Q_0 = \frac{\sqrt{\frac{k}{m}}}{\frac{\gamma}{2m}}. \quad (3)$$

Then

$$\ddot{z} + \frac{2\omega_0}{Q_0}\dot{z} + \omega_0^2 z = \omega_0^2 d + \frac{f}{m}. \quad (4)$$

### 2.2 *Time Invariant Linear System*

All of the control theory in this paper deals with time invariant linear systems. By time invariant we mean that the coefficients ( $m$ ,  $k$ , and  $\gamma$  in our example) of the differential equations don't vary with time.

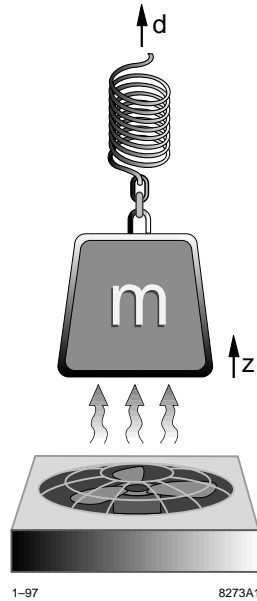


Figure 1: A driven simple harmonic oscillator.

Linear means that there are no terms like  $z^2$  or  $\sin(\dot{z})$  in the differential equations.

Many accelerator problems are naturally time invariant and linear. For those that aren't it is one of the first tasks of the feedback loop designer to find an approximation or coordinate transformation which will make it so. A few examples should make this clear. The transverse feedback system at the CERN SPS [16] must work as the accelerator is ramping up in energy. It is a time dependent problem. In particular, the betatron phase advance between the BPMs and the kicker varies with time. They overcome this problem by processing the data from two BPMs to calculate what the beam position would be at a virtual BPM  $90^\circ$  of betatron phase upstream of the kicker. That is, they preprocess their data in a time dependent fashion to cancel out the time dependence of the situation and present a time invariant problem to the feedback system.

A second example is provided by the beam energy feedback system [10] at the end of the SLAC linear accelerator which measures the energy

error with BPMs and corrects it by adjusting the phases in the RF accelerating sections. While the calculation of the energy error from BPM readings is linear, the calculation of the phases needed to correct this error involves an arc-tangent and is highly nonlinear. This problem is solved by having the feedback loop do the linear calculation of the value of a virtual energy actuator and then in a post processing step doing the nonlinear calculation of the phases needed to get the desired energy gain.

### 2.3 State-Space Notation

State-space notation provides a method of describing a set of  $n^{th}$  order linear differential equations, such as those which occur in feedback systems, in a compact vector-matrix form. This general form is

$$\dot{\mathbf{x}}(t) = \mathbf{F}\mathbf{x}(t) + \mathbf{G}\mathbf{u}(t) + \mathbf{G}_w\mathbf{w}(t), \quad (5)$$

$$\mathbf{y}(t) = \mathbf{H}\mathbf{x}(t) + \mathbf{J}\mathbf{u}(t). \quad (6)$$

It can be written still more compactly as

$$\begin{bmatrix} \dot{\mathbf{x}} \\ \mathbf{y} \end{bmatrix} = \begin{bmatrix} \mathbf{F} & \mathbf{G} & \mathbf{G}_w \\ \mathbf{H} & \mathbf{J} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \mathbf{x} \\ \mathbf{u} \\ \mathbf{w} \end{bmatrix} = \mathbf{S} \begin{bmatrix} \mathbf{x} \\ \mathbf{u} \\ \mathbf{w} \end{bmatrix}. \quad (7)$$

$\mathbf{F}$ ,  $\mathbf{G}$ ,  $\mathbf{G}_w$ ,  $\mathbf{H}$ ,  $\mathbf{J}$ , and their combined form  $\mathbf{S}$  are all constant matrices.  $\mathbf{S}$  is known as the *noise augmented system matrix* and provides a complete description of the physical system. Often the noise terms,  $\mathbf{G}_w$  and  $\mathbf{w}(t)$  are omitted. The resulting matrix is called the *system matrix* and still has the symbol  $\mathbf{S}$ . Some of the literature uses a slightly different notation referring to the  $\mathbf{F}$ ,  $\mathbf{G}$ ,  $\mathbf{H}$ , and  $\mathbf{J}$  matrices as  $\mathbf{A}$ ,  $\mathbf{B}$ ,  $\mathbf{C}$ , and  $\mathbf{D}$ , respectively.

The state vector,  $\mathbf{x}$ , has elements which represent the state of the system. There must be one element for each degree-of-freedom of the system. In this way, the present state of the system is completely described by the state vector. For our simple harmonic oscillator example the states are the position and velocity of the mass,  $z$  and  $\dot{z}$ . Knowing these states and the dynamics of the system (represented by  $\mathbf{F}$ ) one can propagate the undriven system into the future. Note that the choice of states is not unique;  $2z$  and  $\dot{z}/53$  would work as well; indeed, so would  $z - \dot{z}$  and  $z + 29\dot{z}$ . In fact any set of independent linear combinations of the states forms a valid set of states. Of course, in practice it is best to choose as states the simplest physical variables describing the system.



The vector  $\mathbf{u}$  represents the external inputs or actuator settings. For our SHO example, it is the 1-vector  $d$ , the displacement of the top of the spring.

The vector  $\mathbf{w}$  represents the noise or disturbance inputs. Normally white noise goes into these inputs and hence the symbol  $\mathbf{w}$ .

The output vector  $\mathbf{y}$  represents what we measure. For our SHO example it is just  $z$ , the displacement of the mass. Here we are assuming that we have one sensor which measures the position of the mass and that we don't have one which directly measures the velocity. It happens in our example that the observable is exactly equal to one of the states. This is not generally true. It is typically true however that each observable is some linear combination of the states; that is,  $\mathbf{J}$  is typically zero.

This state-space notation is a very important concept for several reasons. First of all, any time invariant linear system can be represented in this form; hence formulas and derivations and computer programs using this notation apply to virtually all manageable feedback problems. Secondly, it is very compact. This applies not only to the formulas, but to the computer programs that use them. Consider for example a steering feedback loop [10] implemented at the SLC. It simultaneously steered electrons and positrons at the end of the linear accelerator. The position and angle of each beam were controlled in both the vertical and horizontal directions. Two pairs of BPMs  $90^\circ$  apart in betatron phase were used and each BPM measured both beams in both directions. This system had a total of 8 states and 16 measurements (observables). The original software which implemented this feedback did not use state-space notation and had DO loops nested 5 deep (particle type, position or angle, direction, BPM pair, and BPM within the pair) that accessed arrays with 3 to 5 dimensions. The system worked beautifully but it was a nightmare to make minor changes. A later implementation using state-space notation [3] just had a vector of 16 measurements, one of 24 states (there were 3 states for each item being controlled because information about previous settings of slow moving correctors had to be kept as states) and some matrix-vector multiplication and vector addition. Due to this notation, it was far easier to modify feedback loops or add new ones as the actual software (nesting of DO loops) didn't change, only the size and contents of matrices and vectors did.

The third reason that state-space notation is important is that it is the notation used in the controls CAD programs. To a large extent, if you can figure out what the states of your system are and write down the system matrix, the CAD programs will do the rest of the job of

designing an optimum feedback loop.

State-space notation is so important that, to make sure it is clear, we will continue on with our SHO example, explicitly write down the system matrix, and then do a small calculation with a CAD program. As mentioned above, the states of the SHO are the position and velocity, so the state vector is

$$\mathbf{x} = \begin{bmatrix} \dot{z} \\ z \end{bmatrix}. \quad (8)$$

Now rearrange Equation 4 and write an obvious identity and include an equation indicating that our observable  $y$  is just the position  $z$ :

$$\begin{aligned} \ddot{z} &= -\frac{2\omega_0}{Q_0}\dot{z} - \omega_0^2 z + \omega_0^2 d + \frac{f}{m} \\ \dot{z} &= \dot{z} \\ y &= z. \end{aligned}$$

Consider the displacement of the top of the spring  $d$  to be the control or actuator and hence associate it with  $\mathbf{u}$ , and the force  $f$  to be the noise input and hence associate it with  $\mathbf{w}$ . Putting this in matrix form we have

$$\begin{bmatrix} \dot{\mathbf{x}} \\ \mathbf{y} \end{bmatrix} = \begin{bmatrix} \dot{z} \\ z \\ y \end{bmatrix} = \begin{bmatrix} -\frac{2\omega_0}{Q_0} & -\omega_0^2 & +\omega_0^2 & \frac{1}{m} \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} \dot{z} \\ z \\ d \\ f \end{bmatrix} = \mathbf{S} \begin{bmatrix} \mathbf{x} \\ \mathbf{u} \\ \mathbf{w} \end{bmatrix}.$$

Using this system matrix, Figure 2 shows the complete MATRIXx program needed to model the response of the SHO to a unit step change in  $d$  and Figure 3 shows the resulting plot.

## 2.4 The Transfer Function

The transfer function is the ratio of output to input of a system and may be used to evaluate its frequency response. If you know the dynamics and hence the system matrix of a system, a controls CAD program can calculate it in a flash. Or, if you have the actual physical system, you can measure the transfer function by exciting the input to the system with a sine wave disturbance, waiting for any transients to die out, and then measuring the amplitude and phase of the sine wave on the output relative to that of the input excitation. A graph of the transfer function gives most of the important information about a system and the change of the transfer function when a feedback loop is turned on

```

FREQ=2;           //Set the resonant frequency to 2 Hz
Q=10;            //Set the quality factor to 10
M=1;            //Set the mass
DELTA_T=1/240;   //Set the sampling time
NUM_SAMP=1200;   //Set number of samples to simulate
OMEGA=FREQ*2*PI  //Convert FREQ to radians per second
//
S = [-2*OMEGA/Q  -OMEGA**2  OMEGA**2  1/M
      1          0          0          0
      0          1          0          0];
NS=2;            //Set number of states
//
// Convert from a continuous to a discrete time system,
// that is from a differential equation to a
// difference equation.
//
S = DISCRETIZE(S,NS,DELTA_T);
//
// Simulate the response to a step in the input
//
[N,Y] = DSTEP(S,NS,NUM_SAMP);

PLOT(N*DELTA_T, Y(:,1), 'XLAB/Time(Sec)')

```

Figure 2: MATRIXx program to plot the response of a simple harmonic oscillator to a step change in its force.

(closed) indicates how well a loop is performing. This graph of the transfer function, on a log-log scale is called a Bode plot. Figure 4 shows the Bode plot for our simple harmonic oscillator example. Note that the vertical scale is in dB. If  $H(\omega)$  is the transfer function then  $20 \log_{10} |H(\omega)|$  is what is plotted. So a value of 20 dB indicates that the amplitude of the output oscillation is a factor of 10 greater than that of the input.

Another description of the transfer function is that it is the response of the system to white noise input. White noise by definition has a flat frequency spectrum. Since we are dealing with linear systems, applying white noise to the system and then Fourier analyzing the output is equivalent to exciting it with one frequency at a time. Our SHO example

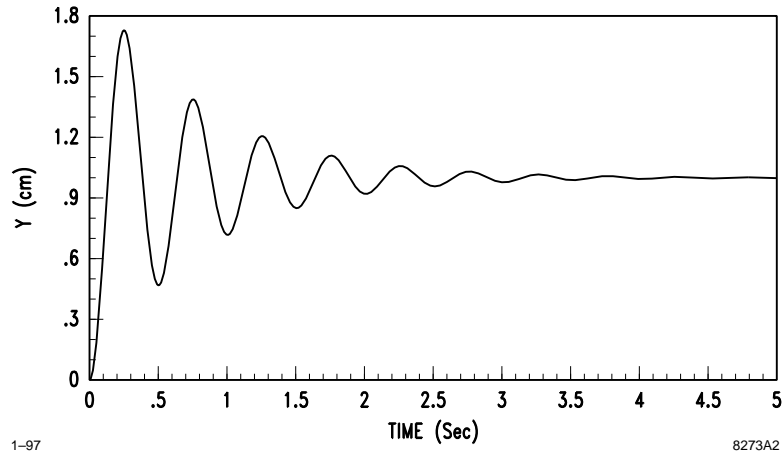


Figure 3: Response of a simple harmonic oscillator to a step change in its input.

will help make this clear. In a sampled system white noise can be simulated by generating a different random number for each sample. Figure 5(a) shows the time response of the position of the mass to this excitation. Figure 5(b) shows the power spectral density, which is just the magnitude of the FFT, of that time response. Finally 5(c) shows the same FFT on a log-log scale. Note that except for the fluctuations (which could be eliminated by averaging several FFTs) this looks just like the Bode plot. For some reason FFTs are normally plotted on a linear scale and Bode plots are always done on a log-log scale, but they are essentially the same thing.

## 2.5 *Feedback and Feed-forward*

The usage of the terms feedback and feed-forward is somewhat confused in the literature, but properly they are quite distinct terms.

*Feedback* is where one measures a quantity with a sensor and based on that reading applies a correction with an actuator. The change of

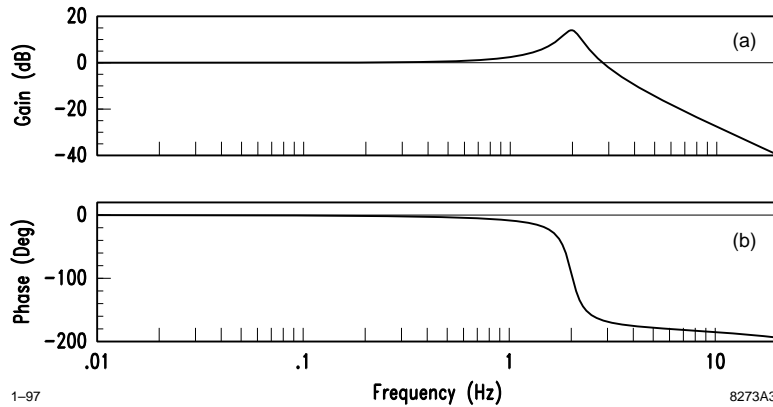


Figure 4: The Bode plot of a simple harmonic oscillator. Part (a) shows the amplitude and (b) the phase.

the actuator has a direct effect on the reading of the sensor. The system is a closed loop. That is, an error in actuator setting will be detected by the sensors in latter measurements and will subsequently be corrected.

There are many examples in the literature of the use of feedback in an accelerator. An example would be the tune feedback system at ELETTRA. [11]. The betatron tune is measured by exciting the beam and finding the peak with a spectrum analyzer connected to a BPM electrode. The correction is made by changing the current in some quadrupoles in the ring. This correction changes the tune of the beam and hence the next measurement.

*Feed-forward* is also where one measures a quantity with a sensor and based on that reading applies a correction with an actuator. However, in this case, the change of the actuator has no direct effect on the reading of the sensor. The system is open loop. That is, an error in the feed-forward calculation and hence the actuator setting will not be detected and corrected in latter measurements with the sensor.

Since it may leave a residual error, feed-forward is usually avoided in favor of feedback. However there are some examples where feed-forward

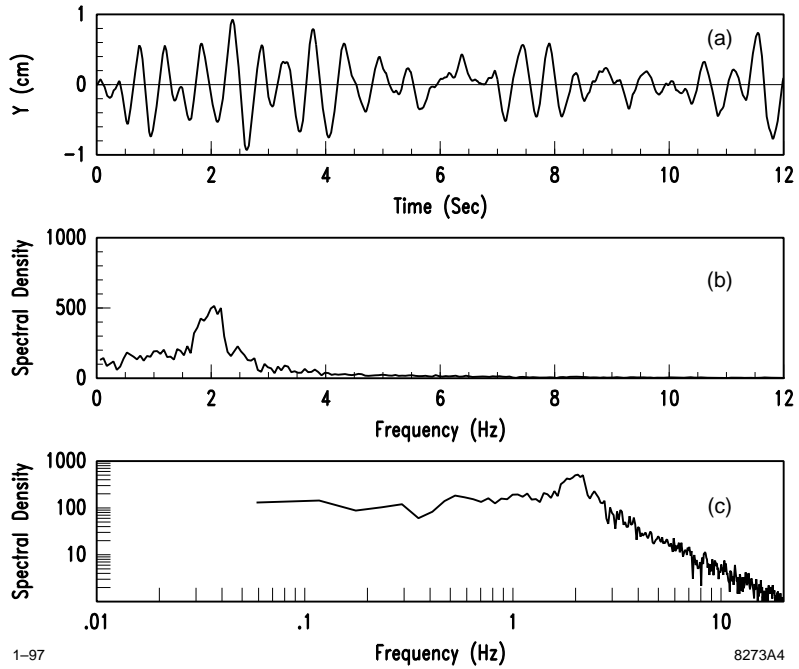


Figure 5: Response of the simple harmonic oscillator to white noise. Part (a) is the time response. Parts (b) and (c) show the power spectral density on linear and log scales.

is useful. For example, suppose you want to control the chromaticity of a ring but its measurement is disruptive to the physics program. During a series of machine development periods you repeatedly measure the chromaticity and discover it is highly correlated with the temperature of the tunnel. You can make a feed-forward system that measures the temperature and sets sextupoles to correct the chromaticity according to your knowledge of this correlation.

A second example of feed-forward, implemented for a different reason, is in use at the SLAC linear collider, SLC [18]. In this case feed-forward is used because it actually works better than feedback could.

The SLC's main linac is pulsed 120 times per second. The particles which will be accelerated are made on preceding pulses and stored in damping rings near the beginning of the linac. The feed-forward system measures the intensities of the bunches while they are stored in the damping rings and uses that information to adjust the phases of the RF in the accelerator to change the total energy gain. This is done to correct for the energy variation caused by the change in beam loading due to the change in beam intensity. This is a feed-forward system because it is not closed loop. The actuator affects the energy but the measurement is an intensity. Even though it is open loop, in this case it is better than (or complementary to) a feedback loop which measures the energy and adjusts the phases. This is because the intensity varies virtually randomly from pulse to pulse. A feedback loop which measures the energy can at best apply the correction on the next pulse. With a nearly random energy variation due to the intensity variation a feedback correction scheme cannot be very effective. The feed-forward system, by using the intensity information from the very pulse that is going to be accelerated, can properly correct for the energy effects of the random pulse to pulse intensity variation. Hence for this case, feed-forward is more effective than feedback.

## 2.6 *SISO and MIMO loops*

A SISO loop is a Single Input Single Output loop. It controls one state with one actuator. If there are multiple sensors they would typically be averaged to get one measurement. An energy or tune loop is typically SISO. A MIMO loop is a Multiple Input Multiple Output loop. It controls multiple states with multiple actuators. A steering loop which controls the position and angle of the beam in a transport line is typically MIMO.

## 2.7 *Analog and Digital Feedback*

The hardware of a feedback loop can be implemented in one of two basic forms.

A digital feedback loop first digitizes the data from the sensors. This is typically done with an analog to digital converter (ADC). The calculations are then done on either a general purpose computer that is part of the control system or on a special purpose computer or digital signal processor (DSP) devoted to the feedback task. The calculated actuator settings are then converted to analog form and applied to the actua-

tor. This conversion is typically done with a digital to analog converter (DAC).

An analog feedback loop doesn't digitize the data. It simply uses filters, amplifiers, analog multipliers, and so on to implement the feedback formulas.

A single, small, simple feedback loop is typically best implemented in the analog form. The hardware for an analog loop is easier (hence faster and cheaper) to design and there is no software to write. Very fast loops (bandwidths of multiple tens of kHz) are almost always done with analog hardware because it is difficult to get enough processing power for such a fast digital loop. To keep the hardware simple, analog loops are typically PID loops designed using classical control theory. (Described in the next chapter.) On the down-side, analog loops are more difficult to change as one learns more about the problem and usually include fewer built in diagnostics.

More complex problems such as a MIMO loop or a loop with significant dynamics where the power of modern control theory (described in the next chapter) is useful lend themselves to a digital implementation. There is also an incentive to go digital at an accelerator where there are many feedback loops of a similar nature [3, 4]. This is particularly true if they are slightly coupled and must interact with each other. Part of this incentive is that while it takes a considerable effort to implement the first digital loop, later ones can be extremely easy to implement.

## 2.8 *Continuous and Discrete Systems*

The decision between analog and digital control also determines whether the system must be handled as continuous or discrete. Necessarily a digital system samples its signals at certain discrete times (normally evenly spaced), performs calculations, and outputs signals to its actuators at discrete times. Such discrete systems are naturally handled with difference equations instead of differential equations. For example the differential equation for the propagation of the state vector taken from Equation 5,

$$\dot{\mathbf{x}}(t) = \mathbf{F}\mathbf{x}(t) + \mathbf{G}\mathbf{u}(t) + \mathbf{G}_w\mathbf{w}(t),$$

changes to the difference equation

$$\mathbf{x}(k+1) = \mathbf{\Phi}\mathbf{x}(k) + \mathbf{\Gamma}\mathbf{u}(k) + \mathbf{\Gamma}_w\mathbf{w}(k) \tag{9}$$

where  $k$  is an integer representing the sample number. Similarly,

$$\mathbf{y}(t) = \mathbf{H}\mathbf{x}(t) + \mathbf{J}\mathbf{u}(t)$$



becomes

$$\mathbf{y}(k) = \mathbf{H}\mathbf{x}(k) + \mathbf{J}\mathbf{u}(k). \quad (10)$$

Note that the differential equation has  $\dot{\mathbf{x}}$  on the left while the difference equation has the new value of  $\mathbf{x}$ , not  $(\mathbf{x}(k+1) - \mathbf{x}(k))/\Delta t$  which would make it more similar to the differential equation. Also note that the matrices  $\mathbf{F}$  and  $\mathbf{G}$  for the continuous case are replaced with  $\Phi$  and  $\Gamma$  for the discrete case and they have different values, and that while the same symbols  $\mathbf{H}$  and  $\mathbf{J}$  are used for both the discrete and continuous cases, they also have different values.

Once again the CAD programs make things easy. As you may have already noticed, the program in Figure 2 includes a call to the function DISCRETIZE which converts a system matrix from continuous to discrete form. Also the function DSTEP is used to get the response to a step change where the “D” stands for discrete. The function STEP would have been used for a continuous system.

## 2.9 *Disturbance Rejection vs. Tracking Reference Changes*

There are many goals or specifications that a feedback loop must be designed to satisfy. Two of the most important ones are disturbance rejection and tracking of reference changes. *Disturbance rejection* refers to keeping the output of a system at a constant value equal to its fixed set-point by counteracting whatever disturbances are making it change. *Tracking of reference changes* refers to getting the output of a system to promptly move when its set-point is changed.

Designing to track reference changes or servo control is what dominates the controls industry. The idea is to have the output of the system closely follow changes of the reference (set-point). For example an airplane should respond promptly and predictably in response to motions of the control-stick, or a robot arm should move in the fashion requested by its program, or a radio antenna should properly track a satellite, or the disk head assembly should move quickly to the proper track on the disk. In these cases it is important to understand the dynamics of the systems: the masses, the moments of inertia, and the flexibility of structural members. Feedback design concentrates on how to compensate for this dynamics to smoothly get the requested motion. Disturbance rejection is often treated as an afterthought. It may be checked at the end to make sure it is adequate, but it isn't an integral part of the design process. In fact, the tracking of reference changes so dominates the con-

trols industry that the  $\Gamma_w \mathbf{w}(k)$  term which describes the disturbance is often omitted or only briefly mentioned in textbooks.

In contrast, tracking is nearly irrelevant in most accelerator applications of feedback. Typically the set-point of a loop is almost never changed and when it is, the response isn't critical. Keeping the beam stable by compensating for unknown disturbances (noise) is the major design criteria of most accelerator feedback loops. Hence it is vitally important to understand the characteristics of the noise and, in fact, to include it in the state space model of the system.

## 2.10 *Modeling the Noise*

In many accelerator feedback applications, the plant (the system to which feedback is supplied) has no, or virtually no dynamics. That is, in the absence of further disturbances, states won't change and the beam won't move because  $\Phi = 0$ . The dynamics of the noise source must be included in the description of the system and calculation of  $\Phi$  in order to properly design an optimal feedback loop. An example will help make this clear.

Consider a steering loop which is to control the position of a beam at the end of a pulsed linear accelerator. It reads the position with a single BPM and controls it with a single fast corrector. In the absence of a feedback loop, there is no dynamic element; there is no spring. If there is a step disturbance, the beam will be offset and will stay offset until there is another disturbance. The system would be described with  $z(k) = \mathbf{w}(k)$  where  $z$  is the beam position, and  $\mathbf{w}$  is the present value of the disturbance. Expressing this in state space form yields  $\Phi = 0$ . For this degenerate case, all of the feedback design methods given in the next section will fail giving no or nonsense results.

Why is this? Well, consider the case where  $\mathbf{w}(k)$  is white noise. That is, for each sample  $k$ ,  $\mathbf{w}(k)$  is an independent random number. For our present example  $z(k) = \mathbf{w}(k)$  is also a random sequence. Hence knowledge of  $z(k), z(k-1) \dots$  yields no information about what  $z(k+1)$  will be and hence no way to correct it with a corrector. The optimum feedback in this case does nothing. On the other hand, consider the case where  $\mathbf{w}(k)$  changes very slowly with  $k$ . In this case, a feedback loop which measured  $z(k)$  and completely corrected it on the next pulse would be quite effective. Since the only thing that differed in these two examples was the frequency spectrum of the noise, you can see that it is critical and must be included in the description of the problem.

This is done by letting the  $\mathbf{w}(k)$  in the  $\Gamma_w \mathbf{w}$  term of Equation 9

always have a white noise spectrum. Elements of the state vector and the  $\Phi$  matrix then describe the dynamics of the noise and hence determine the disturbance spectrum. Described another way, the source of the noise is considered to be part of the plant. Expanding on our linear accelerator steering example will help clarify this important concept.

Perhaps the source of the steering disturbance is the motion of a quadrupole which is on a flimsy stand. Ground motion (which we will consider to be white for this example) moves the bottom of the stand which acts like a spring and moves the quadrupole which moves the beam. The position and velocity of the quadrupole would be states of the system. The equations of motion, which are those for a SHO, could be written down and included in the system matrix.

Perhaps the source of the steering disturbance is the motion of a quadrupole due to temperature variations changing the height of its support. It is known that these temperature variations are very slow compared to the sampling rate of the loop. This can be modeled as white noise going into a low pass filter or integrator. The details of the diurnal variation of the temperature don't need to be included because the loop samples so frequently compared to that time scale. In this case a single state corresponding to the present position of the quadrupole, or equivalently the present position of the beam is used. The dynamics is simply  $z(k+1) = z(k) + w(k)$ .

Perhaps the most likely situation is that the source of the steering disturbance is unknown. In this case we first measure the frequency spectrum of the beam position. Then by trial and error design a filter such that with white noise as input, the measured frequency spectrum is output. Since unknown disturbances are often the sum of many small disturbances and are likely to change with time, it is typically not worthwhile to finely tune the filters to exactly match the measured spectrum. This filter then has states and dynamics associated with it which are included in the description of the plant.

In summary, in plants with little internal dynamics as is typically the case in orbit correction, energy, and tune loops it is vital to include a model of the disturbance as part of the plant. Even loops which have slow actuators (which have their dynamics modeled as part of the plant) must have the disturbance included as part of the plant. For loops where the beam truly has internal dynamics, such as a loop controlling the phase of the beam in a storage ring, that dynamics may dominate, and it may be adequate to use simple white noise for the disturbance.

### 3 FEEDBACK DESIGN FORMALISMS

So far we have covered how to describe the system to be controlled (which in control theory is called the *plant*) using the state-space formalism, how to characterize a system via the transfer function and Bode plot, and the meanings of some feedback related terms. With that background, we now continue on to the meat of the subject, the design of a feedback loop.

The basic task is to design a feedback controller which takes measurements from the plant as inputs and outputs actuator settings which are used to control the plant. Selection of the filters and gains of the controller are critical. Gains that are too large will result in an unstable system where a small perturbation is actually amplified and the system runs away or oscillates. Gains that are too small won't produce the desired disturbance rejection.

There are three formalisms available for feedback controller design: classical, modern, and adaptive. We'll first give a short description of each and then continue on with a section devoted to details of each one.

Classical control theory, as its name implies, is the oldest of the methods. Given a knowledge of the dynamics of the plant, it provides methods to design a feedback loop to meet certain specifications such as the maximum overshoot and settling time allowed on a step change to the input. It provides methods of designing compensating filters and amplifier gains to meet these specifications and ways to check for stability of the loop. In its simplest form, which is what will be covered in the next section, it provides methods to select the three coefficients of a PID (Proportional, Integral, Derivative) loop.

Modern control theory is newer, but has still been around long enough to have textbooks written about it. Given a knowledge of the dynamics of the plant, it provides methods to design a feedback loop to minimize the RMS (Root Mean Square deviation) over time of the states (e.g. beam position or energy) and/or actuator (e.g. corrector) strengths. In a MIMO system one can select relative weights for the contributions of the various states and actuators to the RMS which is minimized.

In practice one is often interested in both the response to a step change of the input and to the RMS of the states. One then uses either classical or modern control theory to optimize one and then plays with the parameters to get an acceptable result for the other. In this way, similar feedback results are likely to result from either design method.

How then does one decide which design method to use? While there

are no absolute rules, here are some rules of thumb:

- If you are planning to use analog hardware and the plant has simple dynamics and is SISO then use classical control theory to design the loop. The equations are simpler and hence easier to implement in analog hardware. In fact, commercial PID controllers are widely available where you just adjust 3 knobs to set the 3 constants.
- If your plant has more complex dynamics or is MIMO or it is really critical to minimize the RMS of the states and the necessary sampling rate is less than a few kHz, then go ahead and do a digital implementation using modern control theory.
- If your situation doesn't fit either of the above, then more thought is necessary.

Adaptive feedback is used for situations where the dynamics of the plant vary with time in an unpredictable fashion. There are many different adaptive methods varying from complete adaptive feedback to adaptive filters used to optimize a feedback loop to adaptive methods that determine the noise spectrum disturbing the plant and then compensate for it. This is a new field and there are very few instances of it actually being applied on an accelerator. Never-the-less, since it is potentially very powerful, we will give a brief description of it in Section 3.3 and give an example of its use in Section 5.2.

### 3.1 *Classical Control Theory*

In this section illustrations are given of some simple applications of classical control theory. Whole textbooks [19, 20] are devoted to this subject and the reader is encouraged to consult them for more details. The coverage here follows that of Franklin, Powell, and Workman [20].

#### 3.1.1 PROPORTIONAL CONTROL

Proportional control is the simplest and most intuitive of all. One just applies a correction proportional to the presently measured error. That is,

$$u(t) = K_p e(t) \tag{11}$$

where  $e(t) \equiv y(t) - r(t)$  is the error to be corrected and  $r(t)$  is the desired reading (reference or set-point) of the sensor  $y$  and  $K_p$  is the proportionality constant or gain.

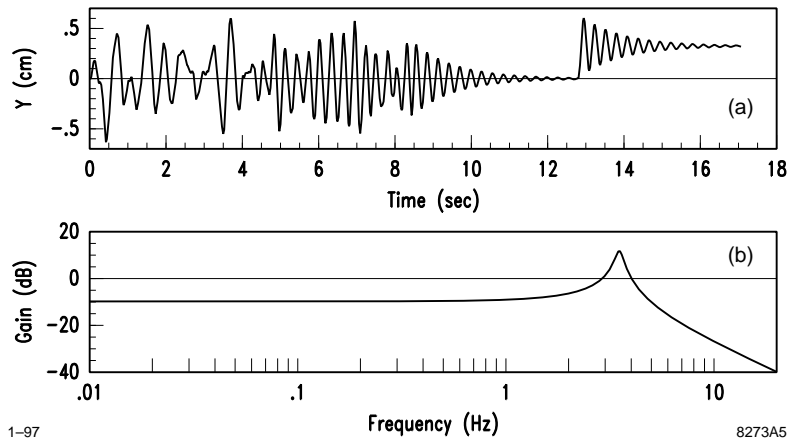


Figure 6: Simulation of a simple harmonic oscillator with proportional feedback. Part (a) shows the time response while (b) shows the frequency response.

The results of a simulation of the SHO with proportional feedback is shown in Figure 6. Part (a) of the figure shows the time response of the system. For the first quarter of the time, the feedback is off (open loop) and the system is excited with white noise. Then the loop is closed (turned on) and the white noise continues for the second quarter of the plot. Then the white noise is turned off so the system damps down during the third quarter of the plot. At that time a sudden step change is made in the disturbance (a sudden DC change in the position of the top of the spring) and the fourth quarter of the plot shows how the system settles down. This same sequence of events will be used in many of the following examples. Part (b) of the figure shows the transfer function or Bode plot.

You can see in both the Bode plot and the time response to white noise with feedback on that the system now has a higher resonant frequency. Also the feedback slightly reduces the RMS of the mass position. However, note that after the step change, the mass does not return to zero as desired. Rather it has a DC offset. The closed loop system has

a poor *DC bias rejection*. This is typically fixed with integral control.

### 3.1.2 INTEGRAL CONTROL

Integral control is used to reduce very low frequency and steady state errors. It applies a correction proportional to the integral over all time of the error. For a continuous system it has the form

$$u(t) = \frac{K_p}{T_I} \int_{t_0}^t e(t) dt . \quad (12)$$

The discrete equivalent changes the actuator by an amount proportional to the present error:

$$u(k) = \frac{K_p T}{T_I} \sum_0^k e(k) = u(k-1) + \frac{K_p T}{T_I} e(k) , \quad (13)$$

where  $T$  is the sampling interval and  $K_p/T_I$  is the gain.

Integral control is particularly useful where the beam has no internal dynamics. A steering loop in a transport line is a good example. The beam may move for many reasons: the gradual drift of a bend supply or temperature changes which displace a quadrupole or the operator adjusting an upstream corrector. There is nothing to automatically bring the beam back to the correct location. An integral feedback loop will change the feedback corrector by an amount proportional to the offset measured in a BPM. Hence it will keep changing it until the offset is zero.

Shown in Figure 7 is a simulation of our SHO example with a proportional and integral (PI) feedback. Note that after the step change the position settles back to zero with no offset. The Bode plot shows the same thing in a different fashion. At low frequencies there is a large negative gain meaning that low frequency oscillations are highly damped.

### 3.1.3 DERIVATIVE CONTROL

As you watch the mass bob up and down on the spring you may be thinking, “It’s simplistic to measure the position and just try to correct it. I know the mass is moving and should take that into account by measuring the velocity and accounting for how much the mass will move by the time my correction is applied.” If so, you’d be inventing derivative control.

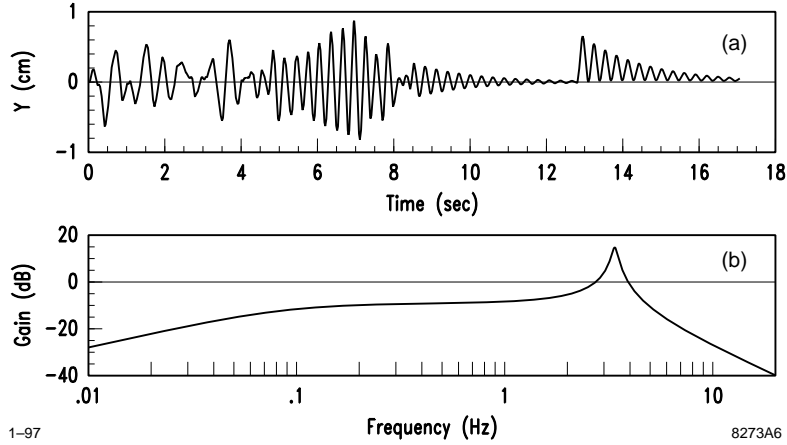


Figure 7: Simulation of a simple harmonic oscillator with PI feedback. Part (a) shows the time response while (b) shows the frequency response.

For continuous systems derivative control has the form

$$u(t) = K_p T_D \dot{e}(t) \quad (14)$$

where  $K_p T_D$  is the gain. Similarly the discrete equation is

$$u(k) = K_p T_D \frac{e(k) - e(k-1)}{T} . \quad (15)$$

### 3.1.4 PID CONTROL

Combining proportional, integral and derivative control gives the PID control law. Its discrete form is

$$u(k) = K_p \left( e(k) + \frac{T}{T_I} \sum_0^k e(k) + T_D \frac{e(k) - e(k-1)}{T} \right) . \quad (16)$$

So now, with the selection of three constants, we have an equation which calculates the actuator setting,  $u(k)$ , based on previous sensor readings of errors,  $e(k)$ . If you know the dynamics of (the differential equations governing) the system there are many ways to calculate these constants



Table 1: PID parameters as determined with Ziegler-Nichols transient response tuning.

	$K_p$	$T_I$	$T_D$
P	$1.0/RL$		
PI	$0.9/RL$	$3L$	
PID	$1.2/RL$	$2L$	$0.5L$

and, in fact, to design more sophisticated compensating filters. Refer to any control theory textbook for more information. However, the dynamics of accelerators are usually quite simple and one can arrive at reasonable constants by a combination of the Ziegler-Nichols PID tuning method and simulation with various trial parameter values using one of the CAD packages.

The procedure for the Ziegler-Nichols tuning method is as follows:

1. Measure the response to a step change of your actuator. This can be done in simulation or on your actual accelerator.
2. Note the time,  $L$ , from when you apply the step change until your sensor reading,  $y(t)$ , has moved 10% of the way to the value it finally settles at.
3. Note the maximum slope,  $R$  on the plot of  $y(t)$  vs. time.
4. Calculate the parameters using Table 1.
5. Simulate the results checking the response to set-point changes, step changes in the disturbance, and disturbances with the frequency spectrum expected in your actual accelerator.
6. Modify parameters and repeat the simulations as needed.

For our SHO example we can get  $R$  and  $L$  from the simulation shown in Figure 3 yielding  $L = 0.04$  sec and  $R = 12$  cm/sec. For a PID loop this gives:  $K_p = 2.08$ ,  $T_I = 0.08$ , and  $T_D = 0.02$ . The resulting simulation with the PID feedback is shown in Figure 8. The simulation actually uses an integral gain one tenth of that obtained with the Ziegler-Nichols formula. Significantly larger gains result in an unstable system. This shows the importance of following steps 5 and 6 of the tuning method.

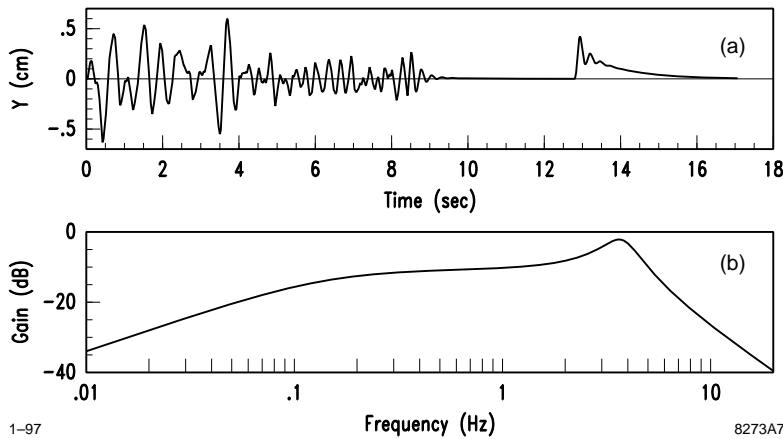


Figure 8: Simulation of a simple harmonic oscillator with PID feedback. Part (a) shows the time response while (b) shows the frequency response.

This large gain reduction was needed because our example oscillator system has a step response quite different from that for which the Ziegler-Nichols method is designed. The integral term causes an instability for the oscillator because at resonance the mass' motion is  $90^\circ$  out of phase with the driving force. The integral of that motion is shifted another  $90^\circ$  giving a total of  $180^\circ$ . So for this case the integral term tends to amplify the motion.

There are other PID tuning methods that can be used. For example, one can excite the real system with a square wave and tune the parameters to get an appropriate transient and DC response [21].

The classical control theory techniques covered in this section are properly applied to many, but not all problems. If you have a SISO feedback problem with simple dynamics, (say no worse than second order) then a PID loop may be the best route to follow. Also, many MIMO problems can be broken down to several SISO problems allowing easy application of these classical techniques. For example a transport line steering loop which corrects the position and angle of the beam with two correctors may be handled as two SISO loops if the correctors are 90 degrees apart and a least squares fit is done to extract the position

and angle from the BPM readings. However, more complex problems, either with complex dynamics (where it becomes difficult to design the proper compensating filters) or MIMO problems which can't be easily split into several SISO problems are better handled with modern control theory. This is covered in the next section.

### 3.2 *Modern Control Theory*

There are several feedback design methods available in the modern control toolbox. One of the most powerful and most commonly used is optimal control using the predictor-corrector formalism [20] described below. We start out with a description of the system to be controlled, the plant, in state-space notation. That is, from our knowledge of the beam dynamics, noise dynamics and actuator responses we have calculated the matrices  $\Phi$ ,  $\Gamma$ ,  $\Gamma_w$ , and  $\mathbf{H}$  and know what the meanings of the elements of the state vector,  $\mathbf{x}$  are. There are two matrix controller equations used in optimal control. The first controller equation is used to estimate the present value of the state vector.

$$\hat{\mathbf{x}}_{k+1} = \Phi \hat{\mathbf{x}}_k + \Gamma \mathbf{u} + \mathbf{L}(\mathbf{y} - \mathbf{H} \hat{\mathbf{x}}_k), \quad (17)$$

where

$\hat{\mathbf{x}}_k$  is the estimate of the state vector on the  $k^{th}$  pulse.

$\Phi$  is the matrix that describes the dynamics of the accelerator and noise models.

$\Gamma$  is the control input matrix. It describes how changes in the actuators should affect the states.

$\mathbf{H}$  is the output matrix. It maps the state vector to the output vector. That is, given an estimate of the states, it gives an estimate of what the sensors should read.

$\mathbf{L}$  is the Kalman filter matrix. Given an error on the estimate of the sensor readings, it applies a correction term to the estimate of the state vector.

The  $\mathbf{L}$  matrix is derived from the other matrices and is designed (via the Linear Quadratic Gaussian method) to minimize the RMS error on the estimate of the state. The CAD packages can calculate  $\mathbf{L}$  given the system matrix, the sensor resolution, and the expected covariance

matrix (typically diagonal) of the noise inputs. It is called a Kalman filter.

While we are not going to derive Equation 17, we can explain it. The first two terms just come from the definition of the  $\Phi$  and  $\Gamma$  matrices in Equation 9. In the absence of disturbances they propagate the state vector over time. The last term provides a correction to the state vector estimate based on the measurements. Note that if the present estimate of the state vector is correct and sensor noise is ignored then  $\mathbf{H}\hat{\mathbf{x}}$  will be equal to  $\mathbf{y}$ , hence the last term will be zero. If the estimate is incorrect, the term is nonzero and  $\mathbf{L}$  is designed to improve the estimate of  $\mathbf{x}$ .

The second controller equation calculates the actuator settings from the estimate of the state vector.

$$\mathbf{u}_{k+1} = \mathbf{K}\hat{\mathbf{x}}_{k+1} + \mathbf{N}\mathbf{r} \quad (18)$$

where

$\mathbf{r}$  is the reference vector which contains set points for the loop.

$\mathbf{N}$  is the controller-reference-input matrix. It maps the reference vector to actuator settings and while it is not part of the system matrix, it is directly derivable from the model of the accelerator.

$\mathbf{K}$  is the gain matrix. It is derived in a manner similar to  $\mathbf{L}$ . It is designed to minimize the RMS over time of a weighted sum of squares of state vector elements and actuator vector elements. Again, it can be calculated by CAD programs given only the system matrix and the weights.

Figure 9 shows the performance of the optimal control feedback on our SHO example. Note that the RMS is better with this feedback than for any of the previous examples. After all, this method is designed to minimize the RMS. In fact, the RMS would be smaller still if the measurement error of 0.1 cm used in the feedback design were smaller. However, note that there is a residual DC offset after the step change in the disturbance. This occurs because the assumed white noise disturbance has very little noise at low frequencies, hence the optimal design doesn't have much rejection at low frequencies. In a real system, if such DC disturbances are expected, the noise model used for the feedback controller design would need to include more low frequency noise such as white noise going into an integrator or low pass filter.

The beauty of optimal control is that even if you have a MIMO system with complex dynamics, all you have to do is describe the system

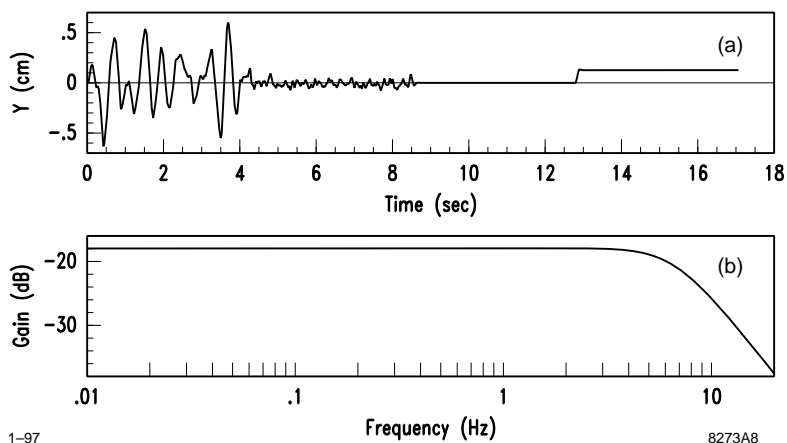


Figure 9: Simulation of a simple harmonic oscillator with optimal feedback. Part (a) shows the time response while (b) shows the frequency response.

in state space form and then a CAD program can do the rest. You end up with a feedback loop guaranteed to minimize the RMS of the states you are concerned about. For complex systems there are some checks that we haven't described which should be done to make sure the system is observable, controllable, and stable. Again the CAD programs do these with just a few more commands.

### 3.3 Adaptive Filters and Control

So far we have been dealing with linear time invariant systems. Most accelerator feedback systems fall into this category or can be cast into it by one of the methods described in Section 2.2. For those that aren't or can't, adaptive techniques may be used.

There are a wide variety of different adaptive techniques suitable for different types of problems. It is not possible to give equations for all of them. Instead, examples will be given of the types of problems that can be solved and references given to papers on applications in accelerators

or to books which describe the technique.

The most general adaptive system is a neural net. It can learn how to handle a nonlinear system and even track changes in it over time. Training the neural net can be tricky and slow, but amazing things can be done if one perseveres. The control of a negative ion source [22] at Los Alamos National Laboratory provides an excellent example of the use of a neural net. Another example uses a neural net (in simulation only) to correct the orbit in storage rings at Brookhaven National Laboratory [23]. The nonlinearity faced here comes from the sextupoles in the ring. A neural net is the only adaptive system which handles nonlinear problems. The rest of the techniques described in this section handle linear time varying problems.

An adaptive noise canceling system [24] can be used when you have a known noise source but the way it couples into the accelerator is unknown or time varying. It is not truly feedback, but it serves many of the same purposes. For example, say the 60 Hz line frequency feeds in through several poorly regulating power supplies and causes the tune of a storage ring to vary. An adaptive noise canceling system would take the 60 Hz line frequency signal and the tune signal, learn the time correlation (including time delays) and produce a signal indicating how to vary the quadrupoles to null out the tune variations correlated with the line frequency. Another example is the use of the signals from several seismometers to null out the beam motion caused by the quadrupole motions generated by the ground motion. In a pulsed accelerator one may be able to do this correction with a much higher bandwidth than that of a standard feedback system whose bandwidth is limited by the pulse rate. A third example, actually implemented on a real accelerator, is described in Section 5.2.

An adaptive method which is truly feedback is the self tuning regulator. Basically one designs a loop with classical control theory techniques. Online, while the loop runs, some historical data is kept. A few of the filter parameters of the controller are then adjusted to optimize the loops simulated performance on the historical data. These adjusted parameters are then used on later data. As time goes on, the filter parameters are continuously adjusted and optimized to track changes in the accelerator or noise spectrum. Barr [25] gives an example of the use of a self tuning regulator for a steering loop.

Many other methods of using adaptive filters and adaptive inverse control are covered in two books [24, 26] on the subject.

## 4 TRADEOFFS

The design of virtually anything requires a series of tradeoffs and compromises. Feedback is no exception. Even though there is a design technique called optimal control, it may not really give the optimum solution to the overall problem. It will do the best possible job of minimizing the RMS that the designer specifies, but the response to a step change, or the robustness of the solution might not be satisfactory. Hence the designer must simulate and test with several parameter sets in order to get a compromise solution that is satisfactory in all regards. The following sections describe some of these tradeoffs.

### 4.1 *Robustness vs. Performance*

It is important to make a loop *robust*, that is, designed to work well even if the dynamics of the real plant are slightly different than those used in the design. For example a steering loop should continue to work well even if the strength of a corrector is a few percent different than the design or the focusing optics are a little off; a loop controlling the phase oscillations in a storage ring must perform well even if the synchrotron tune (the resonant frequency of the SHO) is wrong. Typically this type of robustness is achieved at the expense of reduced bandwidth or gain. By forcing the loop to respond more slowly there is time for it to detect the errors in the dynamics via new sensor readings and then adjust its actuator settings. One should go through the following steps in designing a loop:

- Use the best available knowledge of the plant (accelerator) to design a loop using either classical or modern control techniques.
- Simulate the response of the loop for the plant used in the design and for slight variations of that plant corresponding to the variations expected of the actual accelerator.
- Redesign the loop to improve the robustness. This may be done by reducing the gains of a PID loop, by increasing the sensor noise used in the design of an optimal control loop (this forces the Kalman filter to average over a longer time to average out that noise and hence slows down the response of the loop), or, in a system with a SHO, by decreasing the  $Q$  of the oscillator used for the design. This reduction of  $Q$  will decrease the sensitivity to the exact resonant frequency by increasing the expected frequency spread.

## 4.2 *Performance vs. Dependence on Exact Knowledge of the Noise Spectrum*

The optimum feedback design often depends on the frequency spectrum of the noise that disturbs the beam. This is particularly true for cases where the plant has little or no internal dynamics. Figure 10(a) shows the closed loop frequency response of an optimal controller for the steering loop with virtually no internal dynamics described in Section 2.10. The design was done for a *pink* noise spectrum of the beam position made of white noise plus white noise through a low-pass filter. Note that the loop has a negative gain at low frequencies (i.e. it will decrease the amplitude of incoming low frequency oscillations) but a positive gain at higher frequencies (it will actually excite the beam at these frequencies). Normally one tries to have a high enough sampling rate so that this region of amplification is above the frequency of the noise sources. In a pulsed accelerator this may not be possible (because one can't sample faster than the pulse rate) so modeling of the noise becomes more critical.

Now let's assume we have an additional noise source with a constant frequency of 3 Hz. This is represented as a harmonic oscillator (or a band-pass filter could have been used) and included in the system matrix and state vector. The resulting gain of the newly optimized loop is shown in Figure 10(b). Note the considerably better gain at 3 Hz. Also note the worsened gain on both sides of that frequency. If the beam truly has something exciting it a 3 Hz then this is a good design. However, if the frequency is likely to vary by 50% then this design is not robust and shouldn't be used. So there is the tradeoff: a design with better performance if the disturbance of the beam behaves as planned for in the design vs. one with a poorer performance that is stable for changes in the noise spectrum.

## 4.3 *Actuator Strength vs. Speed and Cost*

Another tradeoff occurs in the selection of the actuator strength. To allow correction of large disturbances, a strong actuator (e.g. corrector magnet) is desirable. On the other side, strong actuators are always more expensive and often slower. For example, a weak corrector magnet can be made with an air core while a stronger one needs an iron core which slows it down. This tradeoff and others like it must be made by the designers looking realistically at the requirements of the loop being designed and using their common sense to make the decisions.



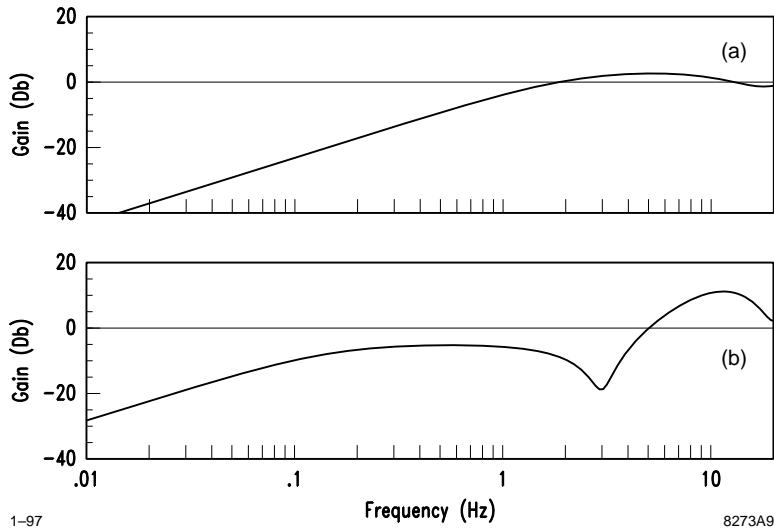


Figure 10: The closed loop frequency response of a steering loop with virtually no internal dynamics. Part (a) shows a design optimized for pink noise and (b) for pink noise with extra noise at 3 Hz.

## 5 MULTIPLE LOOP INTERACTIONS

So far we have concentrated on the design of individual feedback loops. Accelerators often have many feedback loops. These loops may indirectly affect each other because they all act on the same beam. One loop affects the beam which causes the readings of the next loop to change and so on. There are several possible solutions to this problem:

1. Design the loops to be orthogonal so they don't affect each other. For example, two steering loops, one which corrects horizontally and the other vertically can operate independently as long as the x-y coupling of the accelerator is small.
2. Loops running at greatly different speeds are effectively decoupled and hence can operate independently. The slow loop will have a filter which averages over many changes the fast loop makes and

hence reduces the effective coupling while the fast loop can quickly respond to any changes the slow loop makes. To be effectively decoupled in this manner, the bandwidths of loops must differ by over a factor of ten.

3. Loops which are coupled by the accelerator can be designed as nearly independent loops that communicate with each other to coordinate their corrections. An example of this is given in Section 5.2.
4. Corrections at many locations in the accelerator can be combined together to make one large global MIMO loop. In this way the effects of actuators at many locations are directly included in the global feedback design. The next section gives an example of this type of global feedback.

### 5.1 *Example: Global Steering Feedback at APS*

The Advanced Photon Source (APS) at Argonne National Laboratory (ANL) is a 7 GeV positron storage ring designed to produce synchrotron light. Each of the many X-ray beams is stabilized with a local feedback loop. Each loop uses four correctors in the ring to introduce a closed bump in the positron orbit to steer the X-ray beam. By using four correctors to make a closed bump, rather than just two to make a global orbit change that steers the X-ray beam, the designers have applied method 1, making loops orthogonal, to avoid problems with coupled loops.

Unfortunately it is not that simple. Due to eddy currents in the vacuum chamber, magnet calibration errors, and magnet saturation effects, the bumps are not perfectly closed and cause some global orbit error. To cure this error and other sources of global orbit error there is a global orbit feedback system. This system uses 40 BPMs and 38 correctors distributed around the ring to measure and correct the global orbit. This global feedback would move the beam everywhere and hence disturb the X-ray beams. All the loops must run at a high rate so it is not practical to use method 2 to decouple them. Instead they have used method 4 and combined everything together into one large loop [4].

Using the known optics of the accelerator they put together a large response matrix which gives the amount of beam motion at each BPM per unit kick of each of the corrector magnets. A pseudo-inverse of the response matrix was taken using singular value decomposition (SVD).

With this, a set of BPM readings could be converted to the needed corrector settings with a matrix multiply. PID controllers are used.

The distribution and interconnection of the hardware and processing used to implement this global feedback is interesting. There are 20 VME crates distributed around the ring. Each reads its local BPMs and controls its nearby correctors. However, they all need information from all of the other crates. This is accomplished by the use of reflective memories connected by fiber optics. Each crate reads its local BPMs and puts the readings in its local reflective memory. High speed transfers over the optical fibers transport all the BPM data to the reflective memories in all the crates. DSPs in each VME crate then use the information from all the BPMs and do the part of the matrix calculation needed to get the settings for the correctors controlled by that crate. The system runs at 4 kHz.

## 5.2 *Example: Adaptive Cascade at SLAC*

A similar yet subtly different interaction of loops occurs at the SLAC linear collider (SLC) [27]. Here there are seven steering loops placed one after another along the length of the linear accelerator. Each steering loop is a MIMO loop controlling eight states with eight actuators. The states are the position and angle of two beams (electron and positron) in the horizontal and vertical directions. The reason for having so many loops is that it is important to keep the beam centered in the linac to avoid emittance growth due to wake-fields. Each feedback loop uses an optimal controller designed via modern control theory techniques to minimize the RMS of the beam disturbance.

As described so far, the coupling of the loops via the beam is ignored. Consider a sudden disturbance which happens between pulses of the accelerator such as the operator adjusting a corrector. The position and angle will change everywhere downstream on the subsequent pulse. All of the downstream loops will see this disturbance and apply a correction. The net result is an over-correction. The desired action is to have the loop immediately downstream of the disturbance apply the correction. All the other loops should do nothing; i.e. to uncouple the loops and hence avoid the over-correction, each loop should only correct for disturbances which occur between it and the loop immediately upstream.

This is accomplished by having each loop send information to the loop immediately downstream on a fiber optic cable. Hence this is an example of the third method of uncoupling loops: coordinating the ac-

tions of nearly independent loops. In more detail, the following steps are done on each pulse. Figure 11 illustrates the procedure.

1. Read the BPMs.
2. Use the first controller equation (Equation 17) to calculate the estimated states (positions and angles of the beam).
3. Send the eight estimated states to the immediately downstream loop.
4. Receive the estimated states from the immediately upstream loop.
5. Multiply the upstream states by an  $8 \times 8$  matrix of transport matrix elements to calculate how much the upstream disturbances would move the beam at this loop's location.
6. Subtract this upstream disturbance from the estimated states to get an estimate of the disturbance which occurred between the two loops.
7. Use this estimate in the second controller equation (Equation 18) to calculate the corrector settings to fix the disturbance.
8. Output these settings to the correctors.

The extra steps involving communication between adjacent loops allows each loop to separate disturbances that happen immediately upstream from those that occur upstream of the previous loop; hence the loops become uncoupled.

Unfortunately it is not quite that simple. The transport matrix elements used in step 5 vary significantly with time. These variations are caused by uncorrected changes in the energy profile in the accelerator which change the effective focusing strengths of the quadrupoles. These transport matrix element variations must be measured and tracked for the system to work properly. A MIMO adaptive noise canceller is used for this purpose.

The principle involved is quite simple. Consider one particular matrix element of the 64, say the amount the angle changes at loop  $n+1$  when the position at loop  $n$  changes. We could make a plot of  $\text{angle}(n+1)$  vs.  $\text{position}(n)$  and put a point on it for each pulse of the accelerator. If there were no disturbances between the two loops then the natural incoming jitter of the beam would cause the points to lie on a straight line. The slope of that line is the desired transport matrix



online measurement of the transport matrix elements. These measurements are recorded on disk and can be displayed and compared to other changes occurring in the accelerator. Many checks have been made and it is clear that the effects of real variations of the focusing strengths are being measured. The data is used to try to identify and fix the cause of these changes.

## 6 SOME POTENTIAL PROBLEMS AND COUNTERMEASURES

The basic principles of feedback design have been described so far along with a few examples of how they have been implemented at accelerators. Here we will cover a few details which if ignored can cause major problems, but with some forethought can normally be avoided.

### 6.1 *Aliasing*

Aliasing is a problem which only occurs in a digitally implemented feedback system. High frequency signals appear shifted down to lower frequencies because of the finite sampling rate. For example consider a feedback loop that samples a sensor at 100 Hz and assume there is a smooth 101 Hz sine wave on that sensor. Between samples the sine wave will go through 1.01 oscillations. The sensor readings will be exactly the same for this 101 Hz oscillation as they would be for a 1 Hz oscillation. They can not be differentiated. In general, signal frequencies above half the sampling frequency (known as the Nyquist frequency) are aliased down to below that frequency. If nothing is done, a digital feedback loop will respond at the alias frequency which will actually add noise to the beam.

The normal method to avoid the aliasing problem is to use an *anti-aliasing filter*. This is a low pass filter applied to the sensor signal before it is digitized. It is designed to attenuate signals above the Nyquist frequency so they are negligible. This is an easy solution that can be applied to any continuous system. It can be more problematic at a pulsed accelerator where it may be difficult to implement such a low pass filter.

For example (FJ Decker, private communication), the SLC is pulsed at 120 Hz and many of its feedback loops sample the BPMs at 20 Hz. It happens that some of the cooling water pumps have unbalanced impellers which perturb the water at 59 Hz. The water goes through the quadrupoles and makes them vibrate which in turn shakes the beam

at 59 Hz. This is aliased down to 1 Hz which the feedback proceeds to “correct.” It in fact perturbs the beam as was observed by sampling the beam at 120 Hz at a downstream location. The nature of the BPM read-out combined with a lack of CPU power in the microcomputer used for the feedback loop prevented the implementation of an antialiasing filter. The problem has been ameliorated by repairing the pump impellers and by improving the hardware for one loop so it samples and controls at 120 Hz.

## 6.2 *Quantization Noise*

In any feedback system, one must make sure that the sensor noise (error in a sensor reading due to the properties of the sensor and its electronics, not the beam motion) is small enough so it doesn’t significantly degrade the loop performance. Any errors in the sensor readings pass through the feedback controller equations, affect the actuator setting calculated and hence perturb the beam. In a loop designed with optimal control, the sensor noise is one of the inputs to the design. If there is a lot of sensor noise it results in a Kalman filter that averages over many readings, averaging out the noise while slowing down the loop.

In a digital loop there is an additional source of noise. The quantization noise is typically one half of the least significant bit of the ADC. If the ADC doesn’t have enough bits, loop performance will be degraded. This occurred for example, in the transverse feedback at the CERN SPS [16]. The implementation of this feedback is mostly analog. However, a long delay is needed (corresponding to the time it takes the beam to go around the ring) so that the bunch that is kicked will be the same one that was measured. This delay is accomplished by digitizing the signal with an 8 bit ADC and then later putting it back through a DAC. It turns out the single bit noise introduced by the digitization is significantly larger than the analog noise of the system. In fact the digital noise is large enough that the feedback loop would disrupt the proton-antiproton stored beam. Fortunately it is not needed for these runs and is left off. For the running modes where the feedback is needed, the quantization noise is not a problem.

## 6.3 *Actuator Saturation*

To make a feedback loop truly robust it is not sufficient to just design the controller as described in Section 4.1. One must also properly handle exceptions. The two most common exceptions are sensor failure and ac-

tuator saturation. The simple common sense solutions to sensor failure are typically correct so they won't be covered further here. However, the simple common sense solution to actuator saturation is wrong and hence needs further explanation.

By *reaching an actuator limit*, or *actuator saturation* we mean having the feedback controller calculate a desired value for an actuator that can't be physically realized with the available hardware. For example there might be a large orbit offset so a steering feedback calculates that it needs a corrector setting of 5 amps. If the power supply can only produce 4 amps then the actuator has been saturated or its limit has been reached.

The common sense solution to actuator saturation, is to just set the actuator to its maximum value and then continue on with feedback calculations. This is wrong. It results in actuator windup which slows the loop's recovery from the saturation. An example will help make this problem, and its solution, clear.

Consider a steering loop being handled with integral control. Let the gain be -1. From Equation 13 we have

$$u(k) = u(k - 1) - e(k) . \tag{19}$$

If the beam is too low,  $e(k)$  will be negative and the actuator value,  $u$ , will gradually get larger. Let's say that  $u$  reaches the power supply limit of 4 amps while the beam is still too low. If we just set the corrector to 4 amps and continue on with the calculation, the calculated value of  $u$  will continue to increase. Since we aren't changing the corrector any more,  $e$  will remain negative and  $u$  will get quite large. Now if the upstream problem in the accelerator that caused the beam to be so low is fixed, the beam will be too high and  $e$  will be positive. Since  $u$  has gotten so large, i.e. wound up, it will be quite some time before the calculated value gets below 4 amps and we actually change the corrector and fix the beam position. This wind up and the resulting time lag are undesirable. The solution is fairly obvious. The value of  $u$  itself must be limited. Hence Equation 19 becomes

$$u(k) = \text{sat}(u(k - 1) - e(k)) , \tag{20}$$

where the saturation function is defined as

$$\text{sat}(u) = \begin{cases} u_{max} & \text{if } u > u_{max} \\ u_{min} & \text{if } u < u_{min} \\ u & \text{otherwise} \end{cases} . \tag{21}$$



This method of limiting the calculated actuator value also applies to the modern control theory formulation. For a MIMO loop, each element of the actuator vector,  $\mathbf{u}$  is limited individually.

For some loops there may be other considerations. Consider the local steering loops at the APS. Each one uses a four corrector closed bump to adjust the X-ray beam's position and angle. If one corrector saturates so it is moved less than the desired amount, it may be preferable to move the other correctors proportionally less so as to implement a fraction of the desired closed bump rather than to affect the orbit all around the accelerator.

## 7 SUMMARY

The use of feedback in accelerators is becoming more common. Accelerators are now planned which just wouldn't be feasible without their feedback systems. The design of feedback systems has become an integral part of the design and implementation of accelerators. Hence a knowledge of basic control theory and its application to accelerators has become an important tool of the accelerator physicist. This paper has attempted to provide that basic knowledge.

The two formalisms of control theory: classical and modern were introduced. Many examples were given to show how they can be and have been applied to accelerators. For systems which have very little internal dynamics we described how that the dynamics or spectrum of the disturbance that the feedback is trying to correct must be included in the formulation of the problem. Various tradeoffs and potential pitfalls have been discussed.

As electronic technology continues to improve at an incredibly rapid rate and next generation accelerators are built which have still higher requirements than the present ones, feedback systems will only become more powerful and more essential.

## References

- [1] Horowitz P, Hill W. *The Art of Electronics*, pp. 307-390. New York: Cambridge University Press. 2nd ed. (1989)
- [2] Gamp A. *Proc. CERN Accel. School on RF Engineering for Particle Accelerators*, pp. 396-415. Geneva: CERN (1992) ; Boussard D. *Proc CERN Accel. School, 4th, Julich*, pp. 294-322. Geneva: CERN (1991)
- [3] Rouse F, et al. *Nucl. Instrum. Methods*, A316:343-50 (1992)
- [4] Chung Y. See Ref. [33], pp. 1595-7 (1994)
- [5] Chung Y, et al. See Ref. [33], pp. 1592-4 (1994)
- [6] Corbett K, et al. See Ref. [33], pp. 1583-5 (1994); Hettel R et al. See Ref. [33], pp. 1580-2 (1994)
- [7] Martlew BG, et al. See Ref. [33], pp. 1574-6 (1994)
- [8] Nadji A, et al. See Ref. [33], pp. 1571-3 (1994)
- [9] Galayda JN, Chung Y, Hettel RO. In *Synchrotron Radiation Sources*, ed. H Winick, pp. 344-76. Singapore: World Scientific Publishing (1994)
- [10] Abrams GS, et al. *Particle Accelerators* 30:91-6 (1990)
- [11] Carniel A, De Monte R, Tosi A. See Ref. [33], pp. 1601-3 (1994)
- [12] Oxoby G, et al. See Ref. [31], pp. 167-74 (1993)
- [13] Kikutani E, et al. See Ref. [31], pp. 175-8 (1993); Kikutani et al. See Ref. [33], pp. 1613-5 (1994)
- [14] Barry W, et al. See Ref. [32], pp. 2109-11 (1993)
- [15] Hanaoka K, et al. See Ref. [33], pp. 1610-2 (1994)
- [16] Vos L. *Proc. of Accel. Instr. 3rd Annual Workshop, Newport News*, pp. 179-87. New York: American Inst. of Physics (1992)
- [17] Lambertson GR. See Ref. [31], pp. 182-8 (1993)
- [18] Jobe RK, et al. See Ref. [30], pp. 2109-11 (1991)

- [19] Ogata K. *Modern Control Engineering*. Englewood Cliffs, NJ: Prentice Hall Inc. (1970)
- [20] Franklin GF, Powell JD, Workman ML. *Digital Control of Dynamic Systems*. Menlo Park, CA: Addison-Wesley, 2nd ed. (1990)
- [21] Tietze U, Schenke C. *Advanced Electronic Circuits*. New York: Springer-Verlag (1978)
- [22] Mead WC, et al. *Nucl. Instrum. Methods* A352:309-15 (1994)
- [23] Bozoki E, Friedman A. See Ref. [33], pp. 1589-91 (1994)
- [24] Widrow B, Stearns SD. *Adaptive Signal Processing*. Englewood Cliffs, NJ: Prentice Hall Inc. (1985)
- [25] Barr DS. *Proc. of Accel. Instr. Workshop*, pp. 204-14. New York: American Inst. of Physics (1993)
- [26] Widrow B, Walach E. *Adaptive Inverse Control*. Upper Saddle River, NJ: Prentice Hall Inc. (1996)
- [27] Himel T, et al. See Ref. [32], pp.2106-8 (1993)
- [28] Widrow B, et al. *Proc. of the IEEE*. 63(12):1692-716 (1975)
- [29] Himel T. See Ref. [26], pp. 396-408 (1996)
- [30] Lizama L, Chew J. *Proc. of the 1991 Particle Accel. Conf., San Francisco*, Piscataway, NJ: IEEE (1991)
- [31] Kikutani E, Matsuda T. *Proc. of Int. Workshop on B-Factories: Accel. and Expt., Tsukuba*, Tsukuba, Japan: KEK (1993)
- [32] Corneliussen ST. *Proc. of the 1993 Particle Accel. Conf., Washington D.C.*, Piscataway, NJ: IEEE (1993)
- [33] Suller V, Petit-Jean-Genaz C. *Proc. of European Particle Accel. Conf., 4th, London*, River Edge, NJ: World Scientific (1994)