SLAC–PUB–5681
October 1991
(A/I/E)

A DATABASE DRIVEN FAST FEEDBACK SYSTEM FOR THE STANFORD LINEAR COLLIDER*

F. Rouse,◇ S. Castillo,▷ S. Allison, T. Gromme, L. Hendrickson,
T. Himel, K. Krauter, R. Sass, and H. Shoaee
Stanford Linear Accelerator Center
Stanford University, Stanford, CA 94309

Abstract

A new feedback system has been developed that stabilizes the SLC beams at many locations. The feedback loops are designed to sample and correct at the repetition rate of the accelerator. Each loop can be distributed across several INTEL 80386 microprocessors that control the SLC hardware. A new communications system, KISNET, has been developed to pass data between the microprocessors at this rate. The software is written using the state space formalism of digital control theory and is database driven. This allows a new feedback loop to be implemented by setting up the online database and perhaps installing a communications link. Eighteen such loops have now been implemented and this has measurably improved the performance of the accelerator.

Submitted to Nuclear Instruments and Methods A

---

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

◇Current address, University of California Davis, Davis, CA. 95616.

▷Current address, Apple Computer Corporation, Cupertino, CA.

1. Introduction

The SLAC Linear Collider (SLC) is a novel accelerator designed to produce e^+e^- collisions at center-of-mass energies of up to 100 GeV, i.e., around the mass of the neutral intermediate vector boson Z^0. The collisions occur between electrons and positrons produced on every beam crossing and are then thrown away as opposed to being stored for an extended time as in electron-positron storage rings. Before the present project, the SLC had feedback loops to stabilize the energy of the machine [1], the orbit through a set of collimators near the end of the linear accelerator, and one that maintained the beams in collision [2]. These feedback loops are essential to the operation of the SLC. The software for these feedback loops resides on both a VAX 8800 and a series of INTEL 80386 microprocessors (micros). The micros actually control the devices that accelerate and control the beam. The success of the first three feedback loops has led us to redesign the system to allow a more unified and automatic loop specification [3-5] .

We have designed a new system that replaces the specialized software with generic, database driven software. We rely on the SLC database to specify each different loop. This is possible because the action of any feedback loop can be cast into a series of matrix equations in the formalism of digital control theory [6-8] . The SLC database specifies the matrices and describes the vectors on which the matrices act. The database also contains the complete description of what sensors to use (usually beam position monitors) and how to control the actuators (usually magnets) to carry out the changes required to stabilize the loop. We design the matrices and specify the loop in the database, add the hardware for the network linking the different micros in the loop and reboot the micros to start up a new feedback loop in this new system.

The biggest constraint on the new feedback system comes from the topology of the SLC. The accelerator consists of several major instruments: an injector, damping rings, positron target, transport lines, arcs and final focus as shown in

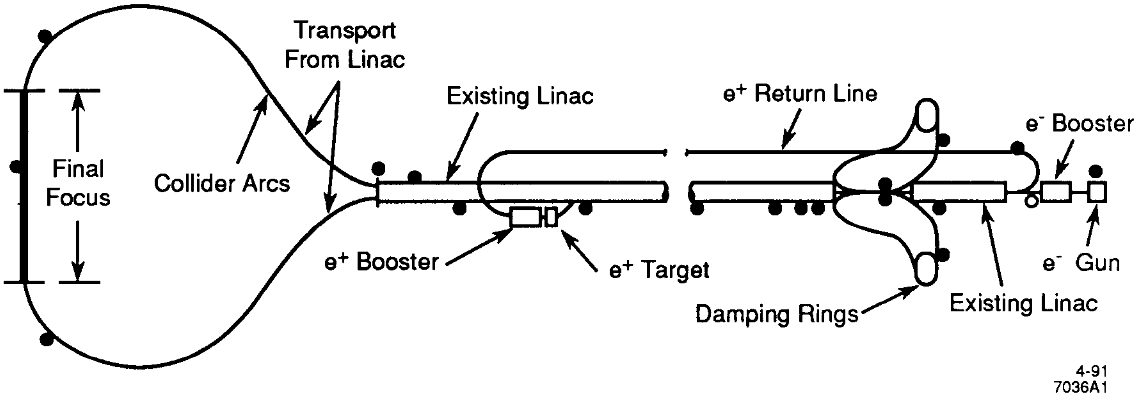

Figure 1. The layout of the SLC. Locations of the presently existing feedback loops are shown with a solid dot.

Figure 1. The major accelerating portion of the accelerator is the LINAC itself. It is divided into 30 *sectors.*

A single micro controls all devices in one geographical region, for example a single transport line, a single sector of the LINAC, a damping ring, etc. Correctors and beam position monitors spread out over several micros are required to measure and control the beam position and angle. Additionally, several feedback loops may need to use devices in the same micro. Hence, a feedback system that has multiple loops executing multiple tasks in a set of micros is required.

Figure 2 shows the basic components needed for one loop. Matrix design is done offline [6]. The VAX orchestrates how each feedback loop works and provides users with timely analysis and status information. The INTEL 80386 microprocessors carry out the processing required for feedback: measurement, computation of the corrections, and control of the appropriate hardware devices. The microprocessors communicate among themselves via a new network called KISNET which is based on the design and hardware of the Advanced Light Source (ALS) [9].

An individual feedback loop may be distributed over several micros. We break feedback into three discrete tasks: measurement, controller and actuator. The

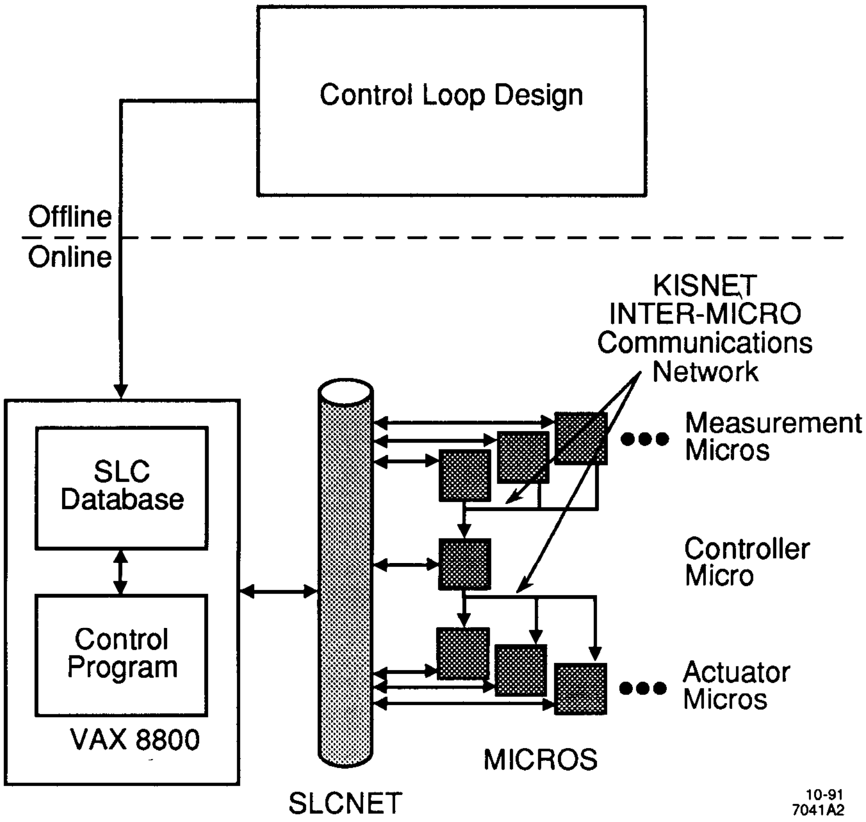

Figure 2. Overview of the components for one feedback loop.

measurement tasks read beam derived data; the controller carries out the matrix arithmetic and determines the next value for the actuators, and the actuator task causes the actuators to be set to the designated values.

1.1 State Space Formalism Used by the Controller

Any continuous linear system can be described by a set of first order differential matrix equations [7]. We can change from continuous time to discrete time by solving this equation and integrating over our sampling intervals. If we had perfect knowledge of the accelerator, we could calculate the exact correction to bring the SLC to any desired state. Unfortunately, this is not possible. Instead we must estimate the state and use the measurements to correct our estimate. The predictor-corrector formalism of state estimation is

$$\hat{\mathbf{x}}(n+1) = \mathbf{\Phi}\hat{\mathbf{x}}(n) + \mathbf{\Gamma}\mathbf{u}(n) + \mathbf{L}(\mathbf{y}(n) - \mathbf{H}\hat{\mathbf{x}}(n)) + \mathbf{M}\mathbf{r} \tag{1}$$

$$\mathbf{u}(n) = -\mathbf{K}\hat{\mathbf{x}}(n) + \mathbf{N}\mathbf{r} \tag{2}$$

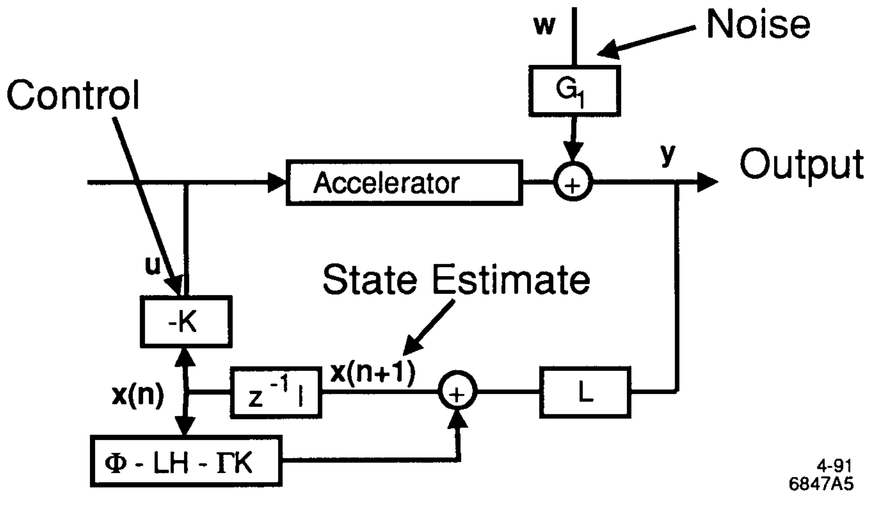

Figure 3. A pictorial representation of the basic predictor-corrector formalism. The operator $\mathbf{z}^{-1}$ represents a delay by one pulse. Omitted from the picture are external references.

where $\hat{\mathbf{x}}$ is the vector of estimated states of the system, $\mathbf{y}$ is the vector of measurements of the system output, $\mathbf{r}$ is the vector of system set points, and $\mathbf{u}$ is a vector of actuation values. Examples of state vector elements include the position and angle of the beam in both the x and y planes, the magnetic field of an actuation magnet, and elements associated with the model of accelerator noise.

The matrices $\mathbf{\Phi}$, $\mathbf{\Gamma}$, and $\mathbf{H}$ represent the system dynamics, account for the state changes caused by the actuators, and connect the current state of the system to the output of the system respectively. The elements of vector $\mathbf{r}$ are the setpoints of the system and the $\mathbf{M}$ and $\mathbf{N}$ matrices can be chosen by the feedback designer [7]. A pictorial representation of the predictor corrector formalism is shown in Figure 3.

The $\mathbf{\Phi}$, $\mathbf{\Gamma}$, and $\mathbf{H}$ matrices come from the model of the SLC. Therefore, we need only concern ourselves with the design of the two matrices $\mathbf{K}$ and $\mathbf{L}$. They are chosen to optimize the response of a feedback loop with respect to response time, overshoot, recovery time, etc., of the loop in response to expected disturbances in the accelerator.

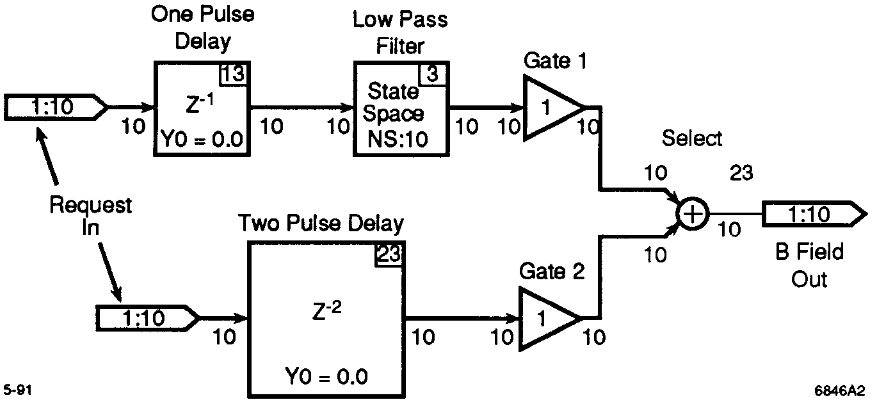

Figure 4. Diagram of actuator dynamics. We model the effect of the magnet as either being a RC time constant due to filtering in the power supply (gate 1) or as a delay of two beam pulses caused by computation and other delays (gate 2). One or the other model is used for each loop, not both.

1.2 Design of the **K** *and* **L** *Matrices*

Control systems are intended to stabilize the operation of dynamical systems such as airplanes, cars, etc. The state of the system at a particular time depends on the state of the system at a previous time. That is, a first or second order differential equation governs the trajectory of the system in time. At first glance, the SLC is not a dynamical system. Accelerator pulses are separated by at least 1/120 sec. Once the beam has gone down the accelerator, nearly all memory of that pulse is lost. The actuator magnets that we use to stabilize the beam do contribute to the dynamics of the accelerator in that they take some time for the magnet current to settle to its requested value. However, this occurs at a time scale fast compared to the 1/120 seconds between pulses [6]. Figure 4 shows our model of the effect of a magnet on the beam.

Yet, we observe disturbances in the SLC beam at very low frequencies. Typically, a considerable amount of power is seen below 1 Hz along with some other noise in frequencies up to 10 Hz or so as shown in Figure 5. These disturbances cause the SLC beam to move. We must model the cause of this dynamics. An ex-

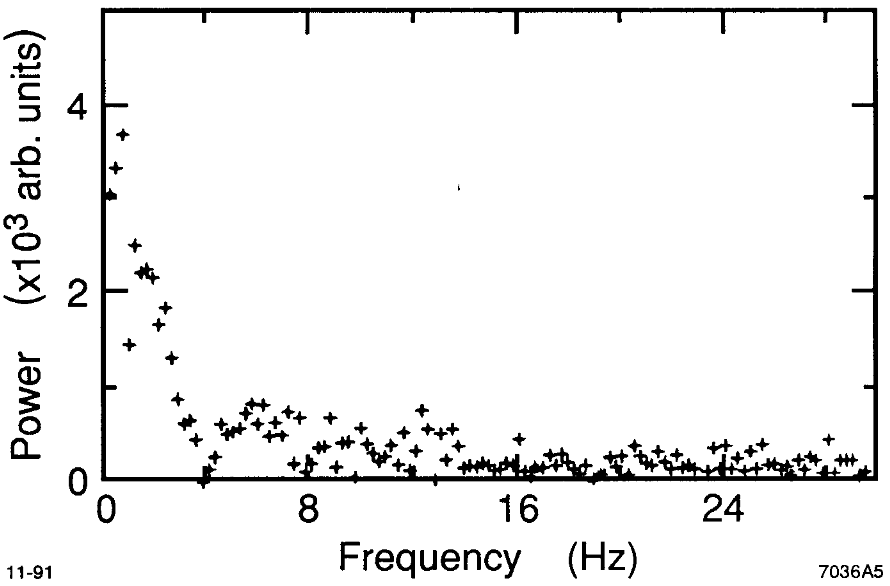

Figure 5. The Fourier transform of the measured position on one Beam Position Monitor in the SLC as a representation of the system noise.

ample could be a slowly oscillating power supply. We must account for the effect of having an ensemble of such power supplies. We adjust the parameters of our model shown in Figure 6 so as to match the observed spectrum of noise actually seen in the collider.

The **L** matrix is then derived via the Linear Quadratic Gaussian method [7]. This method determines the Kalman filter matrix that will minimize the rms error on the state estimate given the expected noise spectrum. Similarly, the **K** matrix is determined so as to minimize the rms of specific state vector elements. The matrices are determined offline by a program built on top of the MatrixX program written by Integrated Systems Incorporated. This program takes parameters such as filter cutoff frequencies, and the transport matrices within the accelerator. The program then produces all of the matrices needed by an individual feedback loop. It calculates and plots several useful diagnostics such as the frequency response of the loop and the response to a step function as shown in Figure 7. The user can then vary the parameters and rerun the simulation until the desired response is obtained. At this point the final matrices are written to the online database.

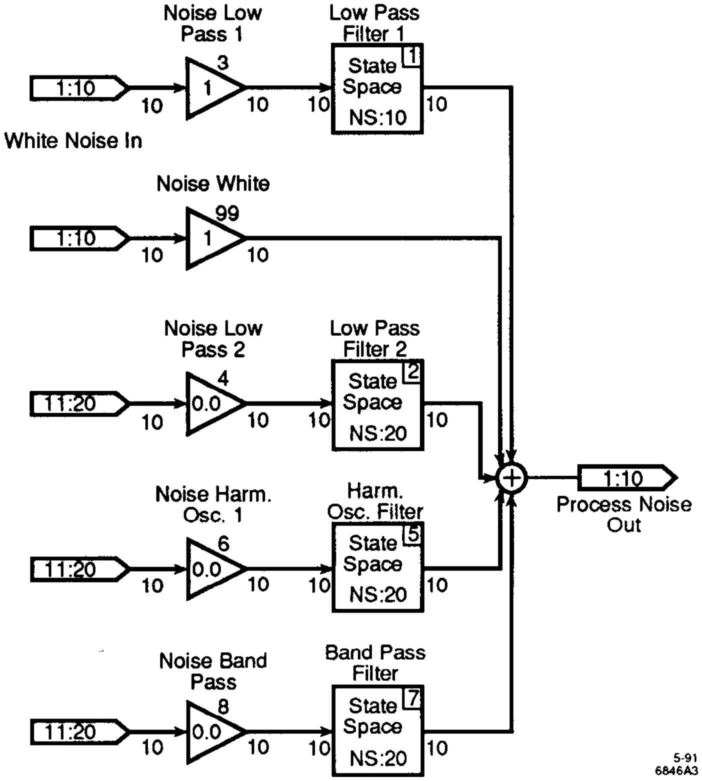

Figure 6. Diagram of beam noise dynamics. Normally only two or three of the gain blocks on the left have nonzero gains. The time constants of the low-pass filters and the frequency and quality factor of the harmonic oscillator filter, can be adjusted to get the modeled noise spectrum to match the measured one.

2. Components of the Feedback System

2.1 VAX Software

A detailed description of the VAX software can be found elsewhere [10]. We only give an overview of the software here.

The VAX is central to the operation of the feedback loops. This is due to the fact that only the VAX has access to the entire SLC database [11]. Each micro only has a copy of the database germane to itself. The VAX, therefore must form the signal routing map between micros and download this map along with other pertinent information at initialization time to the micro.

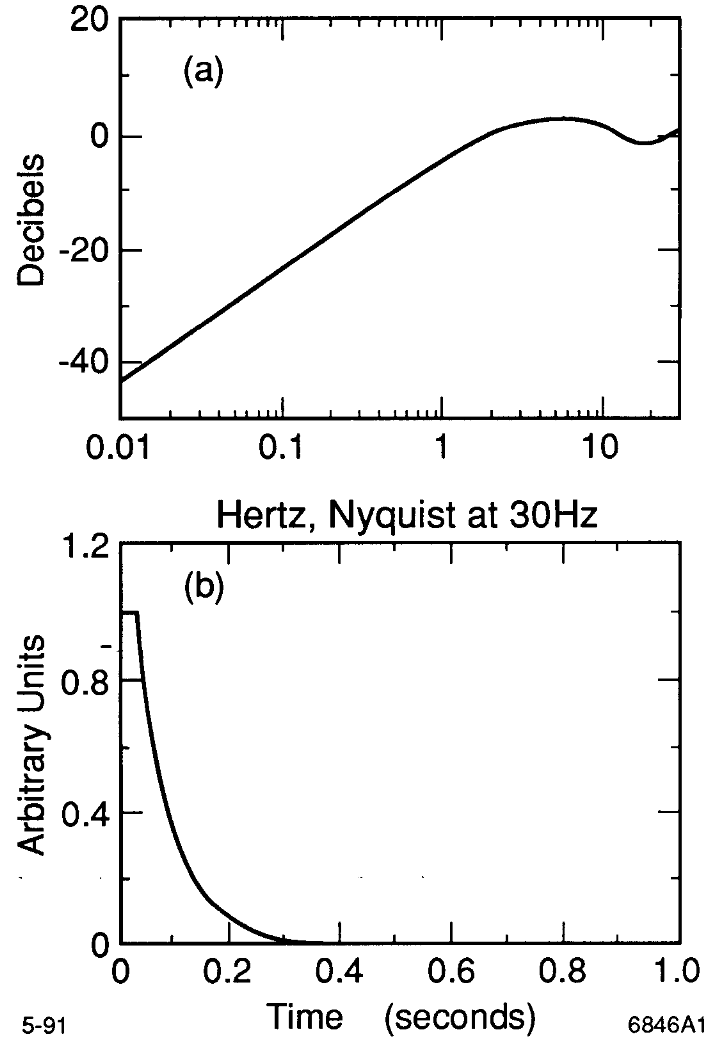

Figure 7. Simulation results: (a) shows frequency response for a typical loop. (b) shows the response to a step change in the incoming states. Typical states recover in 0.1 sec (for a 60 Hz sampling rate).

Additionally, the VAX carries out the functions of data retrieval and display, loop control functions of the system and user initiated actions. The VAX communicates with all micros involved in the system via the bidirectional communications network SLCNET [11,12]. User actions supported by the VAX include loop control and calibration(the measurement of state changes versus actuator changes), diagnostic interventions, display of recent feedback data (measurements, states or actuator settings) by accelerator pulse, entering of loop setpoints and their adjustment via operator controlled knobs, and listing of pertinent loop information.

2.2 MICRO Software

The micro software performs the real time control functions of data acquisition, computation, and device control. A distributed set of micros perform all beam measurements. All measurement data is transmitted to the controller micro which uses the state space formalism detailed in the introduction to compute the required change in the actuator settings to restore the beam. Finally, the actuator settings are transmitted to a series of micros that control the actual devices. A status return is routed back to the controller micro. There are three task types: measurement, controller and actuator. Each feedback loop that has a requirement for a particular task type on this micro is treated as a separate task of that particular type (measurement, controller or actuator). We do this in order to account for multiple loops using a given set of micros for multiple functions. We make a separate task for each function per loop to reduce the problems of keeping track of sources and destinations of data.

For example, if one feedback loop needs measurements from sectors 27 and 28, and controls actuators in sectors 26 and 27 and another feedback loop needs measurements from sectors 28 and 29 and controls actuators in sectors 27 and 28, we would need to create two separate measurement tasks in the micro for sector 28, one measurement task each in sectors 27 and 29, two actuator tasks in sector 28 and one actuator task each in sectors 26 and 28. These example feedback loops along with their KISNET connections are shown in Figure 8 [13].

The purpose of the measurement task (FMES) is to assemble measurement information from all input devices and transmit the values to the controller. FMES communicates with the data acquisition drivers (currently, the Beam Position Monitor (BPM) job) for each class of device. We gather data from all sources on one micro for a feedback loop before transmitting the entire subvector to the controller.

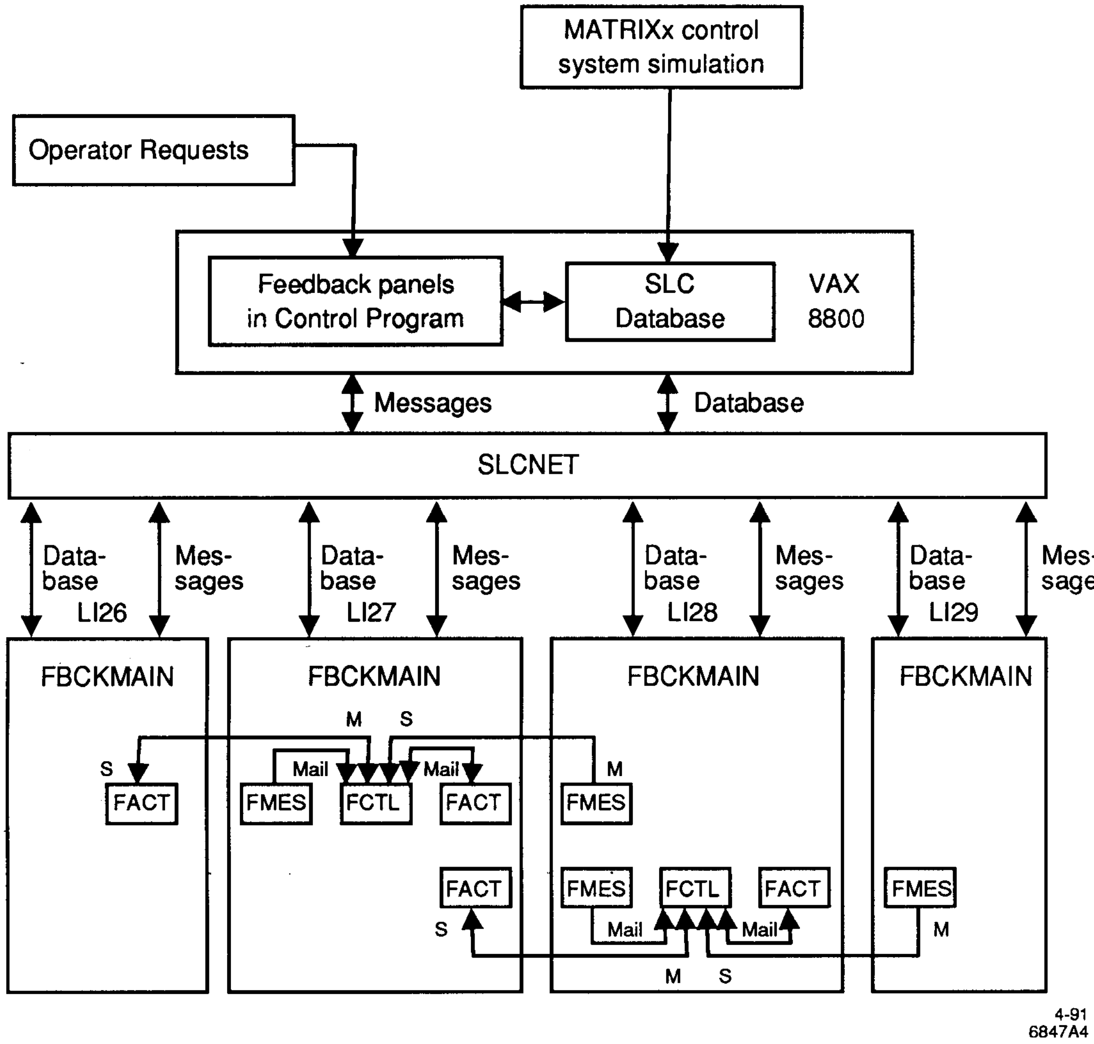

Figure 8. An example of two feedback loops in common micros of the LINAC. Noted are all separate tasks running on each micro. Included in the drawing are the connections both intra-micro (via RMX mailboxes and denoted Mail), and inter-micro via KISNET. KISNET master and slave ports are denoted M and S respectively in the figure.

The controller task (FCTL) waits until all measurement subvectors are assembled before taking action. Once all subvectors from each micro have been received, the controller task implements equation 1 to compute the estimated current state of the machine. It then applies equation 2 to the estimated state to obtain the next actuation settings required to stabilize the machine. The controller is capable of handling non–linear devices such as phase shifters used to control the beam energies by the addition of a short piece of specialized code to the general controller software. We assume the device is linear when designing the feedback loop. The matrices are optimized accordingly and the specialized code converts the linear setting to the actual non–linear device.

The actuator task (FACT) receives the new device settings transmitted by the controller. Each destination micro only receives the subvector of data for devices controlled by the micro. The actuator task then sets the device and reports a status code back to the controller.

2.3 Communications System

A new inter-micro communications network based on the Advanced Light Source (ALS) hardware was built for the feedback system and is described in detail elsewhere [9]. We configure it as a point to point network with a *master* port communicating with a *slave* port. Only one master port can be on any one wire. Multiple slaves can be connected to one master port but we do not operate in this configuration.

The time critical communications, namely measurement to controller and controller to actuator, are implemented by having a master port write to a slave port. Each micro involved in a measurement, therefore, must have a separate master port for each controller to which it must deliver the data. Finally, since only one master can be on a wire, the controller must have one slave port for each measurement micro.

Status values must be returned from the actuators to the controller and are not time critical. Instead of running another wire from each actuator to the controller and therefore creating the necessity of adding one port per actuator micro to the controller, we allow the actuator slaves to write the status values back to the controller master. A master must poll each actuator micro in order to determine if there is status data.

The software is designed to separate the physical transmission of data from higher level functionality. This allows us to change the physical media of transmission (a follow on network) from the conceptual task of transmitting a block of data. For example, some data is passed within the same micro. The lowest level routines use mailboxes provided by the operating system instead of communications ports if the destination is the same micro.

2.4 SLC Database

The database for the feedback system consists of two classes: feedback loop and display information [14]. Feedback loop information includes a loop name, the micros carrying out the measurement, controller and actuator tasks of the loop, the communication links between them, the feedback matrices, and a description of the vectors the matrices act upon. We also specify the state vector that the controller uses to compute the actuator settings. The display information consists of the plot names, windowing for specified plots, and variables.

The matrices are generated offline by modeling the action of the feedback loop along with the model of the accelerator. The results determining the matrices are loaded into the SLC database by the offline program. They are stored in a sparse format (zeros are suppressed).

The vectors must include specific device information. For instance, the measurement and actuation drivers need CAMAC control words and locations in order to read out or set their respective devices. Typically feedback routines only need a pointer to this information. This device information is already part of the SLC database in order to control the accelerator with preexisting applications. Each vector element has a corresponding label that includes the keywords required for unique database access. Finally, the database also describes physical and display units, tolerances, axis labels, etc., for each vector element.

3. Feedback Test Facility

To develop and debug such a large and complex system without adversely impacting accelerator operations, it is important to have a good development and testing environment. The control system simulator, MatrixX, is extremely important in this regard. This software product allows us to tune the model and control matrices in an offline environment. Additionally, we added a second SLC standard 80386 microprocessor to our development system. We also added the standard control electronics for three correctors and three beam position monitors.

A custom "accelerator simulator" chassis was built. This simulator outputs linear combinations of four inputs. Three of the inputs are connected to the DAC outputs of the corrector electronics. The coefficients of the linear combinations are adjusted with potentiometers and are set to values corresponding to transport matrix elements in the accelerator. The fourth input is connected to a signal generator. The three outputs are connected to three channels of BPM electronics.

The signal generator is typically set to generate sine or square waves and simulates a disturbance in the incoming beam. The simulated feedback loop reads the BPMs and adjusts the correctors to suppress the disturbance. This is a very simple loop, but it contains nearly all the complications of a real accelerator loop. We added a loop whose controller was on a different micro than the measurement and actuator tasks to test the KISNET communication.

Once the software works in this development environment it is a small step to get it working on the real accelerator. The final testing and debugging have had very little impact on accelerator operations.

4. Feedback Performance

The new feedback system was installed at the beginning of the April 1991 running cycle. By August 1991, seventeen loops were functioning in the accelerator with the eighteenth implemented in November of 1991. This is a factor of between two and four more loops than originally planned. The demand for more loops and the rapidity with which they were implemented is an indication of how well they worked and the power of having a database driven system.

We currently sample at 20 Hz instead of the beam rate of 120 Hz for two reasons. One is that many loops stabilize both the positrons and electrons but only can make measurements on one beam at a time. Secondly, we are limited by the 386 processor to a 30 Hz sampling rate. We have tested INTEL's new 80486 board with a 33 MHz clock and have achieved a factor of four performance increase for feedback. Hence, for loops that require a 60 Hz sampling rate, we could simply substitute this processor.

Overall, the feedback loops behaved as expected from the simulation. We added an externally adjustable overall loop gain factor. In isolation, all loops could be operated with this gain set to 1.0 without oscillation. However, a string of feedback loops simultaneously measuring and correcting the beam causes overcompensation to upstream perturbations of the accelerator. Because of this we experience overshoot with the ten LINAC loops in a row we currently have running. Hence, we turned down the gain of the loops as a temporary measure pending the implementation of the cascaded system that feeds the state vector forward from upstream to downstream feedback loops. We expect that this will allow us to return the overall gain factor to 1.0 [15].

Figure 9 shows the action of a feedback loop to stabilize the position and angle in both the X and Y planes in the first accelerating section of the linear accelerator. The figure shows the position and angle over a period of 200 seconds in the accelerator in the X plane at injection. We purposefully perturbed the beam by adjusting a corrector and we can see the action of the feedback loop to compensate for it. The positive going response is our initial move of the corrector. The feedback system was able to compensate within $\sim \frac{1}{2}$ sec. About eight seconds later, we restored the corrector to its initial value and again the feedback system

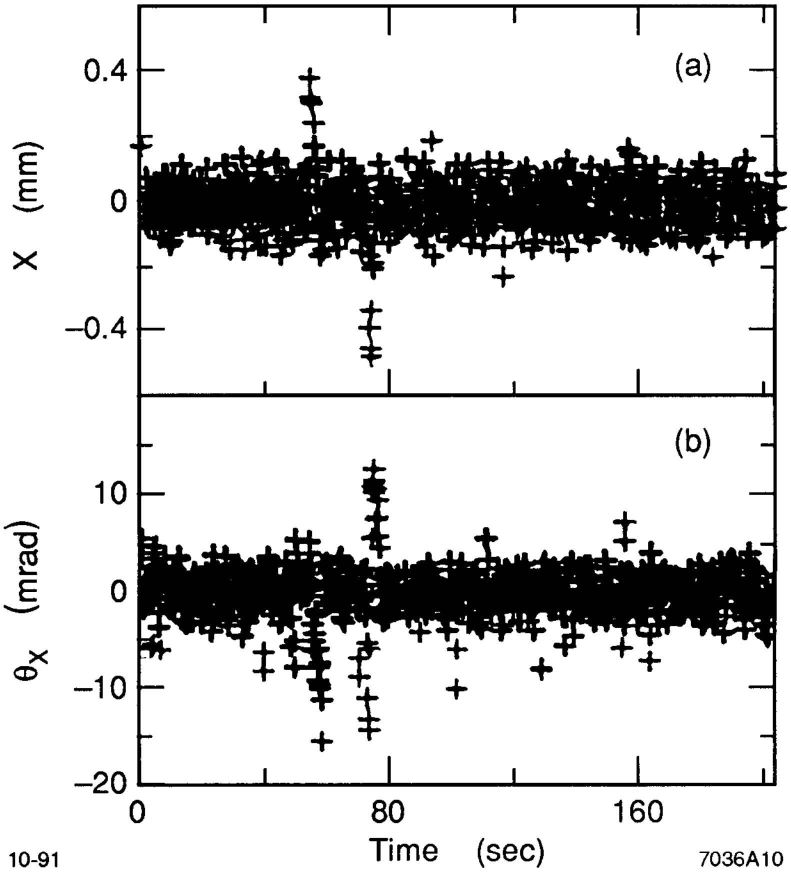

Figure 9. The effect of the LI01 feedback loop on the electron beam. We purposefully moved an upstream corrector not associated with feedback to mimic a random perturbation to the accelerator. This figure shows the response in the X plane for both position and angle of the beam at injection.

compensated for the perturbation. The Y plane position and angle remained nearly constant during the X orbit perturbation.

Figure 10 shows the response of a corrector used in the feedback system for this loop. We see that the value before, during and after the setting of the upstream corrector magnet to initiate the orbit perturbation. The correctors used for the Y plane stabilization remained nearly constant during the X orbit perturbation.

Finally, measurable performance gains were achieved by the collider with feedback. During operations, the operators can control accelerator devices via a series

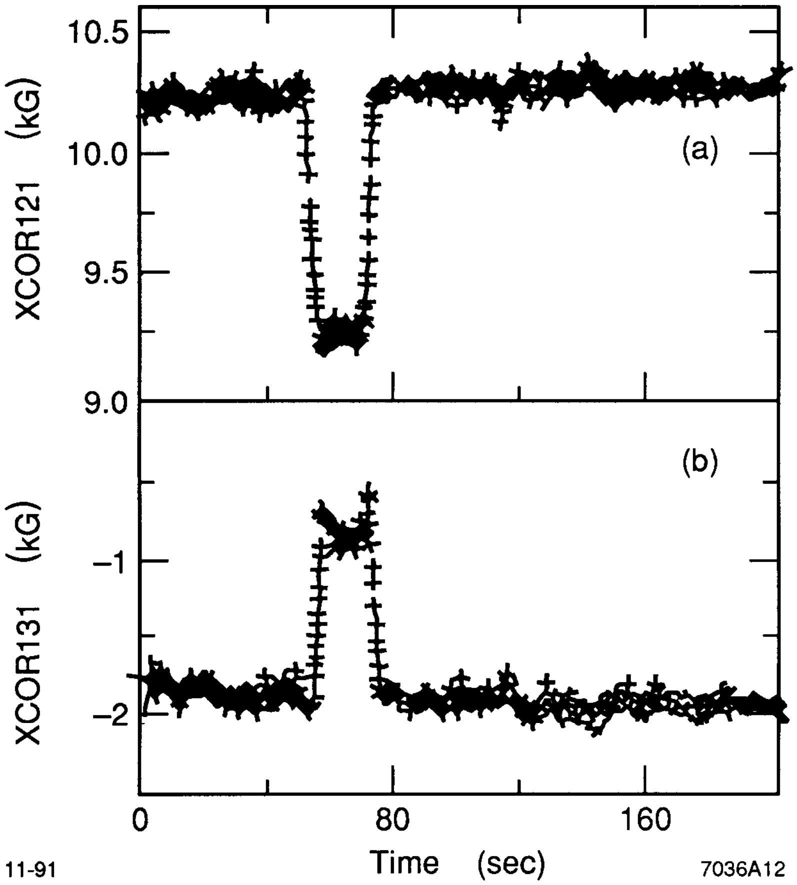

Figure 10. The values of two X plane corrector magnets used by the feedback loop in sector LI01 during the perturbation.

of "knobs." A sample of knob turns is recorded in the collider database. About 15% of the recorded knob turns are related to steering the accelerator and 50% of these knob turns are related to steering the positron return line, the injector of the accelerator and the first accelerating sector of the LINAC. The previous running cycle approximately once every five minutes a knob turn related to these series of devices was made. This running cycle, the rate dropped by a factor of five to once every thirty minutes [16]. The only major change in this portion of the accelerator was the implementation of feedback loops for the injector and the positron return line.

Additionally, a pulse accounting system counts the number of pulses delivered to a number of strategic places in the collider. We can count the number of pulses that were delivered to the interaction point and the number of electrons delivered by the LINAC. The percentage of pulses delivered to the interaction point improved from 30% to 50%. A large portion of this improvement can be attributed to the fast feedback system. Finally, it should be noted that all major operational goals for the SLC were met or exceeded during this running cycle. Much of this success can be attributed to the new fast feedback system.

5. Conclusions

We have described a general feedback system for the Stanford Linear Collider. This feedback system allows us to control the accelerator beam with standard software. We need only make database entries and connect a limited amount of communications hardware to create a new feedback loop anywhere in the machine. A total of eighteen feedback loops have been implemented in this fashion. Typically, we can correct noise frequencies below 2 Hz and respond to step changes in the accelerator within 0.2 sec. Future improvements to increase the speed of computations and the optimal gain of the system are foreseen to improve both of these numbers.

6. Acknowledgements

We would like to thank Gerry Abrams, Eric Soderstrom and Alan Weinstein for developing the first feedback systems that proved the concept of fast feedback for the accelerator. We would also like to thank John Zicker for originating the concept of applying modern control theory to our feedback problem, and we thank Robert McEwen of Integrated Systems Incorporated for his help in applying that theory to our problem. We would also like to thank Robert Hall, Lee Patmore, and Phyllis Grossberg for their efforts on the VAX code.

REFERENCES

[1] G.S. Abrams et al., "Fast Energy and Energy Spectrum Feedback in the SLC Linac," Proc. 1987 IEEE Particle Accelerator Conf., Washington, DC, 1987.

[2] F.R. Rouse et al., "Maintaining Micron Size Beams in Collision at the Stanford Linear Collider," Proc. 1991 IEEE Particle Accelerator Conf., San Francisco, CA, 1991.

[3] Tom Himel, "System Design for Fast Feedback," FFN 000, SLAC, 1990.

[4] F. Rouse, "Current Micro Code," FFN 007, SLAC, 1989.

[5] F. Rouse, "VAX Feedback Specifications," FFN 005, SLAC, 1989

[6] T. Himel et al., "Use of Digital Control Theory State Space Formalism for Feedback at the SLC," Proc. 1991 IEEE Particle Accelerator Conf., San Francisco, CA, 1991.

[7] G. F. Franklin and J. D. Powell, *Digital Control of Dynamic Systems*, Addison-Wesley, 1980.

[8] John Zicker, "Future Possibilities for Feedback on SLC," (talk), 1989.

[9] K. Krauter and D. Nelson, "SLC's Adaptation of the ALS High-Performance Serial Link," Proc. 1991 IEEE Particle Accelerator Conf., San Francisco, CA, 1991.

[10] F. R. Rouse et al., "Design of VAX Software for a Generalized Feedback System," Proc. 1991 IEEE Particle Accelerator Conf., San Francisco, CA, 1991.

[11] Joanne Bogart et al., "SLC Control System Basic Users Guide," SLAC, 1986.

[12] The SLCNET Development Group, "SLCNET Reference Manual," SLAC, 1985.

[13] Bob Sass and Susan Castillo, "Overview of the Top Level Design for the Micro," FFN 002, SLAC, 1990.

[14] F. Rouse, "Feedback Database," FFN 010, SLAC, 1990.

[15] L. Hendrickson et al., "Generalized Fast Feedback System in the SLC," Proc. of International Conf. on Accelerators and Large Experimental Physics Control Systems," Tsukuba, Japan, 1991.

[16] M. Ross, private communication, SLAC.