GENERAL, DATABASE-DRIVEN FAST-FEEDBACK SYSTEM FOR THE STANFORD LINEAR COLLIDER*

F. Rouse,[a] S. Allison, S. Castillo,[b] T. Gromme, B. Hall,[c] L. Hendrickson, T. Himel, K. Krauter, B. Sass, and H. Shoaee
Stanford Linear Accelerator Center, Stanford University, Stanford, CA 94309 USA

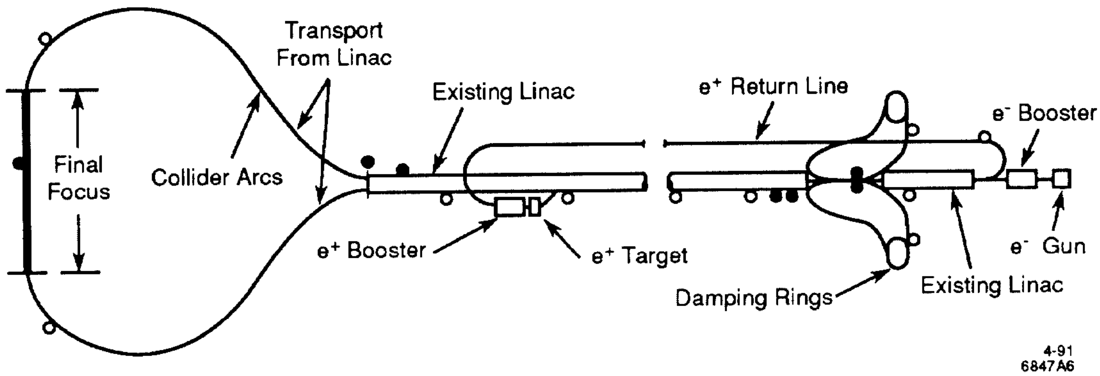

Figure 1. The layout of the SLC. Locations of the presently existing feedback loops are shown with a solid dot. Implementation planned in the next six months are shown with an open dot.

Abstract

A new feedback system has been developed for stabilizing the SLC beams at many locations. The feedback loops are designed to sample and correct at the 60 Hz repetition rate of the accelerator. Each loop can be distributed across several of the standard 80386 microprocessors which control the SLC hardware. A new communications system, KISNet, has been implemented to pass signals between the microprocessors at this rate. The software is written in a general fashion using the state space formalism of digital control theory. This allows a new loop to be implemented by just setting up the online database and perhaps installing a communications link.

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 then thrown away, rather than stored for an extended time as in electron-positron storage rings. Currently, the SLC has feedback loops that stabilize the energy of the machine, stabilize the orbit through a set of collimators near the end of the linear accelerator, and one that maintains the beams in collision. These feedback loops are essential to the operation of the SLC. The software for these feedback loops resides on a VAX 8800 plus a series of INTEL 80386 microprocessors (micros). The micros actually control the devices that accelerate and control the beam. The success of the three feedback loops has led us to redesign the system to allow a more unified and automatic loop specification.

We have replaced the specialized software with generic, database-driven software We rely on the SLC database to specify each 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 [1]. The SLC database specifies the matrices and describes the vectors the matrices act upon. 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 Fig. 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 is required to have multiple loops executing multiple tasks in a set of micros.

Figure 2 shows the basic components needed for one loop. Matrix design is done offline [1]. 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 needed 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) [2].

An individual feedback loop may be distributed over several micros. We break the task of feedback into three discrete tasks: measurement, controller, and actuator. The measurement tasks read beam derived information, the controller carries out the matrix arithmetic and determines the next value for the actuators, and the actuator tasks cause the actuators to be set to the designated values.

*Work supported by Department of Energy contract DE-AC03-76SF00515.
[a] Present address: University of California at Davis.
[b] Present address: Apple Computer Corporation.
[c] Present address: General Electric, Consulting.

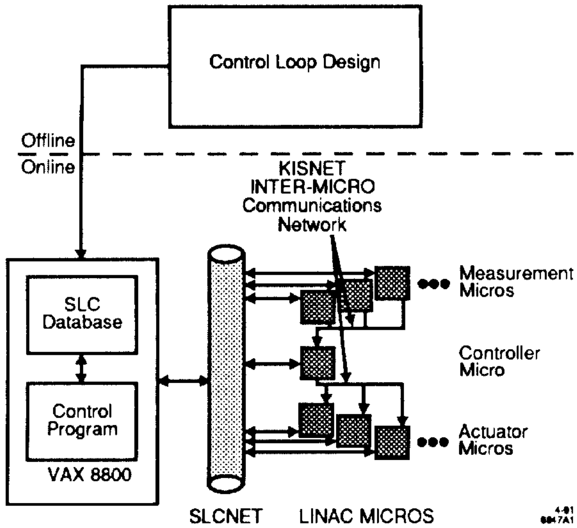

Figure 2. Overview of the components for one feedback loop.

State space formalism used by the controller

Any continuous linear system can be described by a set of first order differential matrix equations [1]. 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

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

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

where $\hat{\mathbf{x}}$ is the vector of estimated states of the system, $\mathbf{y}$ is the vector of measurements of the system output and $\mathbf{u}$ is a vector of actuation values. 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 [1]. A pictorial representation of the predictor corrector formalism is shown in Fig. 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.

COMPONENTS OF THE FEEDBACK SYSTEM

VAX software

A detailed description of the VAX software can be found elsewhere [3]. We only give an overview of the software here.

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

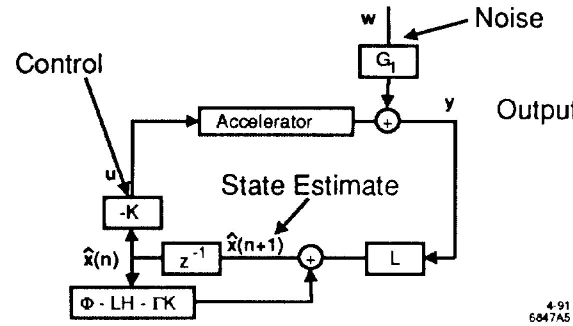

Figure 3. A pictorial representation of the basic predictor-corrector formalism. The operator z^{-1} represents a delay by one pulse. Omitted from the picture are external references.

Additionally, the VAX carries out the functions of information 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 bi-directional communications network, SLCNet. User actions supported by the VAX include loop control and calibration, diagnostic interventions, display of recent feedback data (measurements, states, or actuator settings) by accelerator pulse, and listing of pertinent loop information.

MICRO software

We view the feedback loop as consisting of beam measurements being carried out on a series of micros, with the information being transmitted to the controller micro. The controller micro then uses the state space formalism detailed in the previous section to compute the required 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. On any one micro, one feedback job called FBCKMAIN is created that oversees all three task types: a 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).

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 Fig. 4.

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. Various classes of devices are handled, in addition to beam position monitors. We gather all information from all sources for a feedback loop on one micro before transmitting the entire subvector to the controller.

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 Eq. (1) to compute the estimated current state of the machine. It then applies Eq. (2) to the estimated state to obtain the next actuator settings required to stabilize the machine. The controller is capable

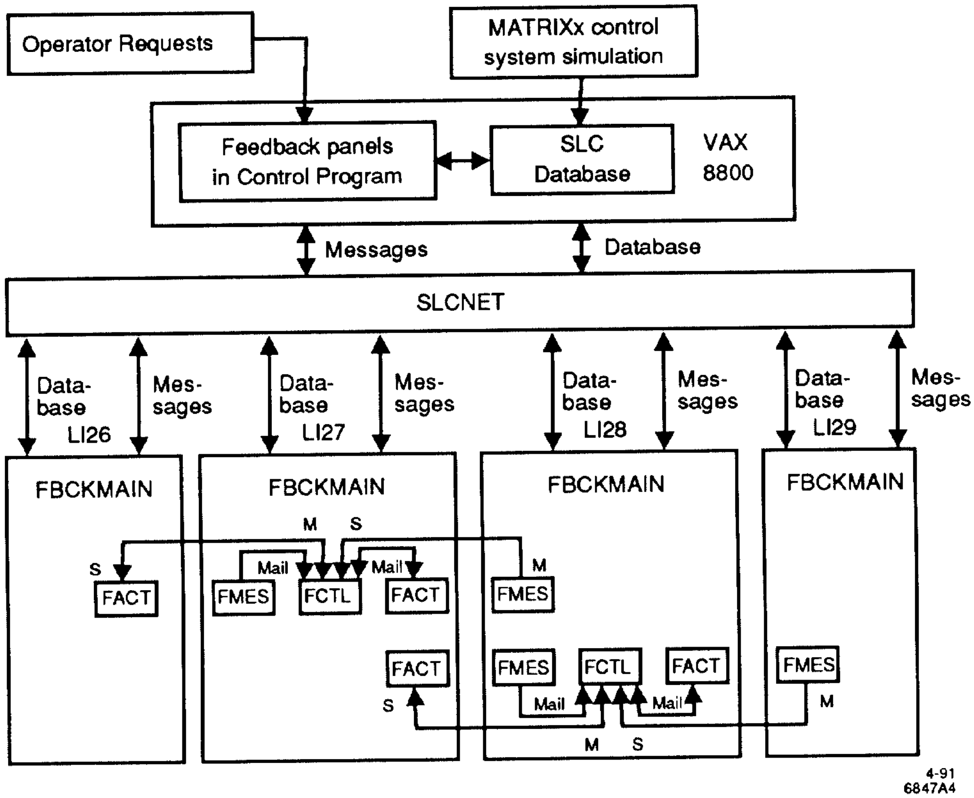

Figure 4. An example of two feedback loops in common micros of the LINAC. Each separate fast-task box corresponds to a separate task under the main task FBCKMAIN. Included in the drawing are the connections both intramicro (via RMX mailboxs and denoted Mail), and intermicro via KISNet. KISNet master and slave ports are denoted M and S respectively in the figure.

of handling nonlinear devices such as phase shifters used to control the beam energies. We expect that in the future this calculation will include state information fed forward from upstream feedback loops.

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

Communications system

A new intermicro communications network based on the Advanced Light Source(ALS) hardware was built for the feedback system and is described in detail elsewhere [2]. 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.

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 must therefore have a separate master port for each controller to which it must deliver the information. Finally, since only one master can be on a wire, the controller must have one slave port for each measurement micro.

Status information must be returned from the actuators to the controller. This information is 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 information back to the controller master. A master must poll each actuator micro in order to even 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 information 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.

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.

ACKNOWLEDGMENTS

We thank John Zicker for his early work on this problem. We also thank Lee Patmore and Phyllis Grossberg for their efforts on the VAX code.

REFERENCES

[1] 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.

[2] K. Krauter and D. Nelson, "SLC's Adaptation of the ALS High Performance Serial Link," ibid.

[3] F.R. Rouse et al., "Design of VAX software for a Generalized Feedback System," ibid.