MODEL-BASED EXPERT SYSTEMS FOR LINAC COMPUTER CONTROLS*

MARTIN J. LEE

Stanford Linear Accelerator Center,
Stanford University, Stanford, California 94309

ABSTRACT

The use of machine modeling and beam simulation programs for the control of accelerator operation has become standard practice. The success of a model-based control operation depends on how the parameter to be controlled is measured, how the measured data is analyzed, how the result of the analysis is interpreted, and how a solution is implemented. There is considerable interest in applying expert systems technology that can automate all of these processes. The design of an expert system to control the beam trajectory in linear accelerators will be discussed as an illustration of this approach.

MODEL-BASED CONTROL PROGRAMS

The design of linear accelerators requires the use of computer programs in a variety of roles. At the most basic level, the designers use these programs to estimate the effects of various beamline configurations on observable beam parameters of interest with the goal of finding an "ideal" configuration that yields beam parameters meeting specific design requirements. At another level, the designers use the programs to compute the effects of configuration errors on the observable beam parameters and to find correction schemes that compensate such effects. At the highest level, these programs, used as components of the monitoring and control systems, allow the beamline operator to efficiently achieve and maintain proper machine operation.

The primary function of the modeling programs is to compute a number of machine functions and beam properties at various positions for given beamline configurations. Here, "configuration" is defined as the location and parameter values for each beamline element. In general, the beam properties are computed from the machine functions. As an example, the beam trajectory is a typical beam property that is computed from the machine function called the Transport Matrix which specifies the relationship between the beam coordinates at one point with those at another. For linear accelerators the beam properties also depend on the launch conditions, the beam parameter values at the entrance of the beam line. Typical launch parameters are the beam coordinates and momenta at the beginning of the LINAC.

While it is possible for the designer to lay the plans for the ideal machine, it is generally not possible to build it exactly. During construction, errors in fabrication, calibration, or installation of the elements can be made. These unknown errors can cause the as-built machine to be different from the ideal machine conceived in the design. Unless these errors can be found and corrected, it may be impossible to bring the machine performance to the ideal design specifications. During commissioning, when the machine is first tested with a beam, modeling and simulation programs can be used either to find the element errors or to correct their effects. Using a simulation program, it is possible to find the errors by studying the effects produced by suspected errors. After the errors are found, they can be corrected. However, in some cases, correcting the errors in the machine elements may not be possible or practical. For such cases, the model can be altered to represent the as-built machine. I believe that one of the important goals in commissioning is to obtain a realistic model of the machine such that the beam simulation from the model matches the measured values. Once a model has been shown to represent the machine, it can be used for the control of the beam during machine operation. Many of the look-and-adjust operations such as changing the LINAC lattice, controlling the beam launch conditions, and correcting the trajectory errors can be done using the model-based beam simulation programs. A schematic diagram showing the relationships between the users, the machine, the beam, and the model-based control system is shown in Fig. 1.

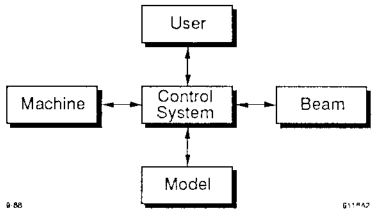

Fig. 1. A schematic diagram showing the relationships between the user and the model-based control system.

A SYSTEMATIC APPROACH TO AUTOMATION

For the past several years, several laboratories have been developing software packages for building control systems for any accelerator. The goal of these projects is to build a toolkit that can reduce the computer programming efforts required to build a state-of-the-art model-based control system. Tools exist, at various laboratories, to build a manual model-based control system (e.g., database building tool, graphic and menu building tools, beam line modeling tools, etc.). Recently, there has been considerable interest in the development of automation application tools. One of the necessary tools for an automatic control system is a model-based expert system for making beam parameter changes.

In general, every manual look-and-adjust operation to change the beam parameter values can be described in terms of the four basic processes: (1) Look–the operator looks at the beam parameter values (the observables) to decide on what changes to make, (2) Analyze–the operator uses the modeling or simulation code to calculate the change in the controllable values that will change the observable values to what he wants, (3) Interpret–the operator makes a decision on whether to implement the solution based on the result of the prediction. (4) Adjust–the operator makes an adjustment of controllables to make the predicted changes. In practice, the user may choose

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

to repeat these four processes until the observables reached the desired values.

After a given manual look-and-adjust procedure has been used successfully, it is possible to use it to control the beam parameters automatically. The conversion from manual to automated operation can be done by replacing the operator's decisions in each of the processes with a set of rules (an expert system). An expert system is a set of rules that an expert uses in making decisions. These rules can be written in any programming language. I will call this way of Applying the Intelligence of an expert the "real AI". To automate an existing manual procedure, the first step is to restructure it so that each of the four processes can be automated independently. A possible solution is to use the database as a buffer between each of the processes as shown schematically in Fig. 2. Each process gets its input and output parameter values from the modeling database. I will illustrate the rules for automating the four basic processes in a trajectory correction procedure in the next four sections.

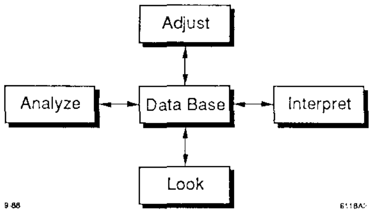

Fig. 2. A schematic diagram showing the four basic processes in a model-based beam parameter change operation.

AUTOMATING THE ADJUST PROCESS

The rules for the Adjust Process deal with adjusting the values of the controllables intelligently. The Adjust Process involves changing the set-point values of the controllables in order to make changes in the strength of the observables to some desired values. It also involves the conversion of strength values to set-point values. For the case of trajectory correction, the controllables will be a specific set of correctors. The set-point value of a corrector is given in volts and the strength value is in milliradians. The Adjust Process first converts the strength values to the set-point values before it can make the adjustment.

In general, a mathematical function (the model) is used to represent the relationship between the set-point value and the element strength value. For example, the strength of a quadrupole magnet can be expressed as a polynomial function of the set-point value. The modeling parameter of the quadrupole magnet are the coefficients of the polynomial expansion (the magnet calibration data) stored in the database of the control system.

In some applications, it is required that the strength of a specific set of elements must be changed simultaneously. For these cases, the rate of change of the set point values must be adjusted according to the desired rate of change of the strength values. In order to do this, the model is used to compute the values of the set-point for each desired incremental change in strength values.

In particular, for cases in which the relationship between the set-point values and the strength values is highly nonlinear, it is necessary to check the readback value of the element strength at every incremental change of the set-point values. This can be done by computing the strength values (from the readback set-point values) and comparing them with the desired strength values. The expert system can be used to monitor their differences in order to identify and report any failures in the automated adjust process. It is not sufficient to monitor the difference between the desired set-point values and the readback set-point values.

AUTOMATING THE LOOK PROCESS

The rules for the Look Process deal with how to measure the values of the observables intelligently. I will illustrate the general features of rules for making beam trajectory measurements. For this case, the Look Process consists of: (1) converting the beam signals to the beam positions; (2) repeating the measurement to reduce the effects of the noise in the beam signals.

In practice, the measured signal values are generally given in volts and the position values are given in millimeters. A mathematical function is usually used to represent the relationship between the measured signal values and the position values. For example, the BPM signal value can be expressed as a polynomial function of the beam position values. The modeling parameters are the coefficients in the polynomial expansion that are determined from the calibration data. It is possible to write a simple expert system (a set of "If-Then-Else" rules) that checks the model prediction, i.e., the result of converting BPM signal values to position values. It is necessary to check this conversion for cases when the relationship between the signal values and the position values is highly nonlinear and the calibration data points are sparse. The result of this calculation should be included automatically as a measure of the measurement accuracy.

It is also possible to write a set of simple rules that filters out the bad data points. For example, if the beam position is given as the average value over several BPM scans (Num-Scans), a measure of the accuracy is given by the standard deviation value (Sigma). If the Sigma value is greater than the acceptable value (Max-Sigma), the average value should be ignored. Else, throw away all scans with the position value deviation from the average value greater than the acceptable value (Max-Deviation). The expert system should also decide to make new scans or to calculate the new average value. These simple rules can be used to reduce the noise in the measurement due to fluctuations. A more sophisticated expert system can be developed to automatically determine the optimal values for the system parameters Num-Scans, Max-Sigma, and Max-Deviation.

AUTOMATING THE ANALYSIS PROCESS

The rules for this process are used to solve for the unknowns intelligently. The unknown can either be the strength change of the controllables, or the values of the errors in the controllables or observables. This process involves analyzing the measured observable values to find the unknowns. In general, the analysis process consists of: (1) Calculate the values of the controllables to obtain some desired change in the observables; (2) Check the solution to be sure that it is acceptable (meeting additional constraints or requirements).

Consider the Analysis Process for trajectory correction in the SLAC LINAC as an example. The observables are the measured change in the trajectory values at the BPMs. The controllables are the strength of the dipole correctors along the two-mile beam line. The calculation may be done by minimizing the objective function (measured trajectory change minus predicted trajectory change from the correctors) squared by varying the values of the corrector strength. In order for the solution to be useful, the objective function value has to be less than an acceptable value (Max-Objective) and the predicted corrector strength has to be less than the maximum value (Max-Correction).

One possible way this analysis can be done is to use an optimization program that finds a global optimum solution. In this calculation, the user has to specify the candidates to be used as correctors. The program finds automatically the values of corrector strength that minimize the objective function subjected to the constraint condition: the strength of the corrector is less than Max-Correction. The use of a Global Optimum (GO) program to find such a solution is straight-forward providing that the value of the objective function is less than Max-Objective. In the event that the objective function is greater than Max-Objective, it will be necessary to decide what to do (pick a different set of correctors or modify some of the constraint conditions). An expert system can make such decisions automatically.

AUTOMATING THE INTERPRET PROCESS

Rules are used to evaluate the results after implementing the solution given by the GO program. The results can be the difference between the predicted change and the measured change in the values of the observables. This process involves comparing these changes to decide what action will be needed. It requires a lot of expertise (the real AI). In general, the Interpret Process is needed when there is more than one possible option to be chosen. In this process, the various options are compared and the optimal one is chosen. One of the options is to decide when to quit.

For example, in the case of trajectory correction, when one or more correctors reaches its Max-Correction limit there could be more than one way to solve for an acceptable solution. It would be necessary to choose which is the optimal way. Moreover, if the Analysis Process failed to find any acceptable solutions, it would be necessary to find the causes of the problem rather than to correct the effects. Some of the possible causes of trajectory errors are the launch errors or errors in the beam line elements. It is possible to develop an expert system to make hypotheses on what the causes of the problems are and to suggest the tests to be made to solve the problems.

AN EXPERT SYSTEM FOR TRAJECTORY ERROR ANALYSIS

As a prototype project, two expert systems for the analysis of trajectory errors have been developed to find errors automatically. The first expert system can not be used at SLAC because of the way it was written as will be described below. The second expert system can be used to find errors automatically (off-line) in the elements at the SLAC LINAC such as position or strength errors in quadrupole magnets, and off-set errors in the beam position monitors. Since this is a prototype system, it has not been implemented (on-line) into the SLC control program. I would like to describe my experiences in its development.

In this prototype study, the first expert system was written in LISP using a commercially available expert system building shell. It was done as a graduate thesis (MS) by a student at KSL, the Knowledge Systems Laboratory at Stanford. The time needed in this prototype development was about two-man years. Half of the time was spent developing an error simulation program to find the errors in the beam line elements that may cause the measured trajectory errors (in collaboration with the Real Time Systems Group at LBL). The rest of the time was spent developing the rules manually and implementing them into an expert system program.

The rules were developed using simulated trajectory errors from a simulation program. It was my job as an "expert" to find the errors in the beam line elements. A simulated trajectory was calculated with a known set of errors in the machine elements using the beam simulation program. I was given the simulated trajectory errors and asked to find the machine element errors. Here is the learning process that was used: (1) This trajectory error was analyzed manually using the error simulation program. (2) Different assumptions were made on the possible locations of the element errors. (3) For each assumption, an optimization program was used to find the values of the element errors that produced a best fitted solution to the given simulated trajectory errors. (4) By repeating this trial-and-error procedure and looking for a systematic way to find the unknown element errors, a set of rules was developed. A block diagram depicting the manual/automatic error analysis procedure is shown in Fig. 3.

Unfortunately, because the expert system shell was unavailable at SLAC, it was not possible to use this program to analyze the actual trajectory errors at SLC. Furthermore, the rules that I had discovered require searching for the unknown over a very large search space so it was impractical to apply manually. Thus, it was necessary for me to come up with a different procedure.

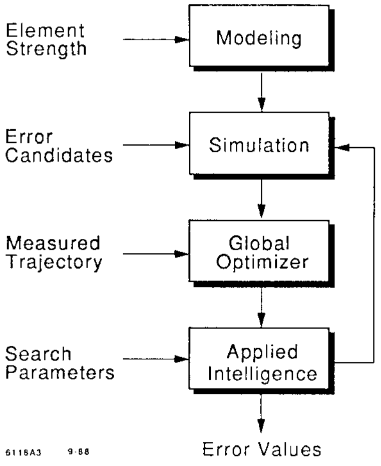

Fig. 3. A block diagram to show the rule learning process using modeling and simulation programs.

From my experience in the commissioning of the SLAC LINAC, I noticed that most of the errors were large mistakes. If the mistake is small, the correction elements can be used to correct its effects. What I wanted for commissioning was a set of rules to find large localized mistakes. Once I understood the nature of the problem, it became obvious that I should first look for regions that have no mistakes (the error-free regions) since these regions are large and easy to find. The search of the errors will be done over the small subregions (small search space) that are outside of the large error-free regions. I was able to develop two sets of rules: one for finding the error-free regions, and one for finding the errors.

In order to implement these rules at SLAC, they were written in FORTRAN. Unfortunately, because the rules were written in FORTRAN, they are difficult to modify. Because of this difficulty, upgrading of these rules has been kept to a minimum. It is possible for the operator to use this program as an off-line procedure to find errors today. Whenever changes in the beam trajectory have been noted during operation, the expert system can be used to find the causes of these changes.

AUTOMATING THE BEAM TRANSFER PROCESS

In general, a linear accelerator is only one of the subsystems of the entire machine. For example, the SLC consists of several subsystems: an injector, damping rings, two-mile LINAC, arcs, and a final focus section. Any error in a machine element can affect the beam properties downstream. As the control program corrects the errors in one subsystem, the beam properties in all the subsystems downstream are affected. In order to operate the entire machine automatically, an end-to-end model-based expert system is needed. The conceptual development of a model-based control system for automatic beam parameter changes amongst subsystems is considered in this section.

In the design stage, each subsystem can be designed independently of other subsystems. Because different design requirements are imposed on the beam parameters in a particular subsystem, different modeling and simulation codes may be used in their design. The only requirement imposed is that the beam parameter values at the exit of a given subsystem match those at the entrance of the subsystem downstream.

It is conceivable to design the end-to-end model-based control system to consist of model-based expert system modules. One module is assigned to operate a specific subsystem (see

Fig. 2). Each module has its own database which contains both the values of the beam line parameters (locations and strengths of each element) and beam parameters (size, shape, centroid of the beam at the entrance and exit of every beam line element).

Since the beam is the only connection between the subsystems, the beam parameter values at the exit of one subsystem are used for the beam parameter values at the entrance to the next subsystem. The conversion of the beam parameter values between any adjacent subsystem can be handled by a Beam Transfer Processor. Each Beam Transfer Processor gets its input from the database of the next module as illustrated in Fig. 4. An expert system can be used in each of the Beam Transfer Processes to check the continuity of the beam parameters between subsystems.

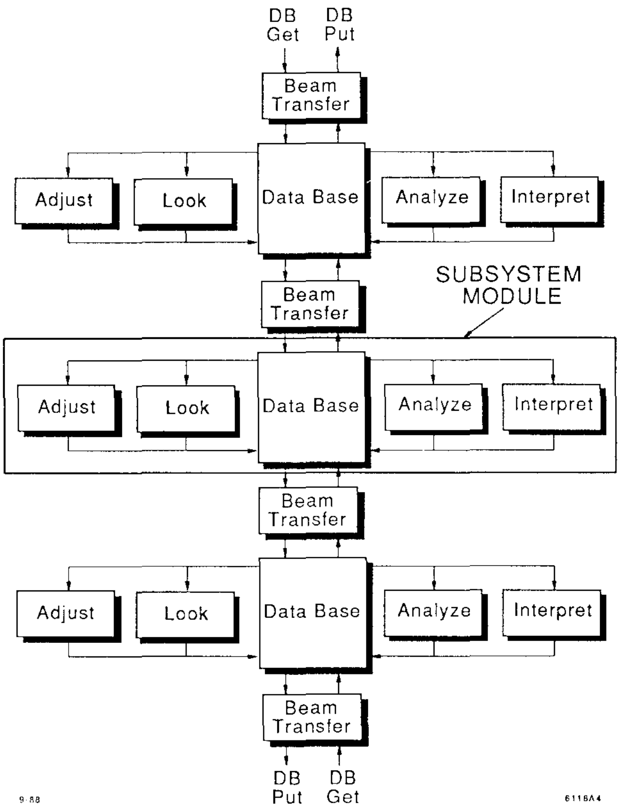

Fig. 4. A schematic diagram showing how the model-based control modules for the subsystems are connected.

A NEW DESIGN APPROACH

Until now, I have been describing a systematic way to convert a conventional system for an existing machine to an automated model-based control system. Are we always going to take this "morning-after" approach even for future machines? From my own experience in commissioning (the two-mile LINAC, SPEAR, PEP and SLC) over the past twenty years, I have come to the conclusion that the degree of difficulty in machine commissioning or beam recovery is directly related to the availability of tools. An automated beam parameter change procedure is one of the more useful tools. I would like to consider four levels of automation. Table 1 gives the definition of the these levels and the degrees of comfort in commissioning and operation of the machine. I believe that the following five steps should be taken during the construction of new machines: (1) develop an end-to-end simulation program from the design codes, (2) design the correction schemes using these programs, (3) learn how to operate the machine systematically (rules) using simulated beam data, (4) develop expert systems from these rules and (5) implement the expert systems into the control program. The automation of accelerator control for future machines should not be an afterthought.

Table 1. Definition of the four levels of automation and the degree of comfort for each level.

	Level of Automation			
	1	2	3	4
Interpret	Manual	Manual	Manual	Auto
Analyze	Manual	Manual	Auto	Auto
Look	Manual	Auto	Auto	Auto
Adjust	Auto	Auto	Auto	Auto
	Degree of Comfort			
Control System	Old	Conventional	Modern	Future
Commissioning	Very Difficult	Difficult	Hard	Moderate
Operation	Difficult	Hard	Moderate	Easy

An example of an afterthought is shown here in a memo one of the operation engineers (R. Iverson) wrote to me recently: "A new tool that would be very helpful is an automated injector bunching package. Whenever the SLC beam current is changed significantly, the injector parameters must be adjusted to re-optimize electron bunching. This can be very time consuming because there are so many parameters to adjust and monitor. One can imagine a routine which looks at the quality of the electron bunching as the operators do. After this data is analyzed, the injector parameters can be adjusted based on an online model. Then a new observation could be made to check that the bunching is optimized. The fact that the injector is tuned by a consistent set of rules and that manual tuning and correction is very time consuming due to the number of parameters involved, makes the injector bunching process a good candidate for automation."

ACKNOWLEDGEMENTS

I would like to express my thanks to Bob Huff for many stimulating discussions and useful suggestions on the subject of this paper, and Stephen Kleban for reading over the manuscript many times and suggesting a number of relevant changes. I also would like to express my appreciation to Yadunath Zambre, Herbert Weidner and Joyce Lee for editing assistance.