Framework for a General Purpose, Intelligent Control System for Particle Accelerators

R. T. Westervelt, W. B. Klein

Vista Control System Inc., Los Alamos, New Mexico 87544

and

G. Luger

University of New Mexico, Albuquerque, New Mexico 87131

ABSTRACT

Tuning and controlling particle accelerators is time consuming and expensive. Inherently nonlinear, the control problem is one to which conventional methods cannot satisfactorily be applied. Advanced information technologies such as expert systems and neural networks have been applied separately to the problem, with isolated success. Few, if any, of these advanced information technologies have been applied for general use or in a manner useful to multiple accelerator installations. We discuss results of coupling neural network and expert systems technology to solve several standard accelerator tuning problems based on realistic simulations. We also examine the effectiveness of additional heuristic search techniques such as genetic algorithms. Finally, we show the integration of this hybrid AI system with an existing general-purpose control system.

PROJECT OVERVIEW

The goal of this project is to develop a very flexible, intelligent controller that can reduce the tuning time for a particle accelerator and can develop "better" tunes than are now achieved by human operators. Additionally, the intelligent controller should maintain the tune with smaller deviations than are currently exhibited. Various approaches have been taken to automate the control of accelerators [1,2,3], with varied degrees of success. Generally, most effort has been directed toward solving problems for a particular facility and little effort has been directed to developing more general solutions applicable to a number of different accelerator facilities. This paper reports the early status of this project after the first phase of the research.

The architectural framework for the controller is an expert system that guides more specialized controllers based on the state of the system and the tuning goal. We developed a realistic simulation environment to test the controller operation. We have examined several types of controllers, including back propagation neural networks, fuzzy logic controller, analytic based tuning by the expert system, and genetic algorithm tuners.

Steering, a standard zero-order problem, is one of the initial tasks of a beamline tuner. We first considered the basic situation in which steering was controlled by two steering magnets (SMs) separated by some distance. Two beam position monitors (BPMs) downstream of the steerers

monitored the effect of the steerers. Steering must take into account beamline alignment, electronic offset and drift, and downstream tuning requirements. In general, there is jitter in the initial beam coordinates with some frequency. Beam-source mechanical and electrical variation causes this jitter, which limits steering accuracy.

Another basic element of beam transport is the periodic line for focusing. Beam root mean square (rms) sizes are measured on profile monitors (PMs), which directly measure intensity distribution. These PMs, usually wire scanners, contain inherent inaccuracies due to beam fluctuation during measurement and component error. The relationship between quadrupole settings and beam profile is nonlinear, making accurate tuning difficult even for human operators.

ACCELERATOR SIMULATION

We began by developing a program to simulate the steering scenario for initial prototyping of the intelligent controller. This test program was a linear model for beam deflection through the steering magnets. We ignored higher order effects of the steering magnets. The noise effects of beam variations, magnet variations, and measurement variations in the BPMs were all included in the simulation. Additionally, we included random device failures as an option in the simulation.

We interfaced TRANSPORT [4], a standard accelerator modeling program, to Vsystem, our commercial software product for developing control systems to provide a rich simulation environment. We modified TRANSPORT by adding input types to relate card deck elements to Vsystem database channels. We then modified TRANSPORT to automatically recalculate the simulation when any input parameters changed. Automatic recalculation effectively simulates the real world response of an accelerator.

We added realistic noise and error effects by filtering and varying the data as the data was stored in the Vsystem database. Random gaussian noise was added to data signals for monitoring devices. Noise characteristics were configurable from the Vsystem database. Time dependent device behavior was also included in the simulation.

EXPERT SYSTEM

An expert system is a computer system that can help solve complex, real-world problems. Expert systems use large bodies of facts and procedures gathered from experts. These

facts and procedures are usually domain specific knowledge gathered from real-world experience and not necessarily equation-based constraints or foundational theorems. Expert systems use rules and facts to reason and make decisions, often using imprecise or incomplete information. Most expert systems also have the ability to explain their reasoning and decisions.

Our design used an expert system at the top level for reasoning and control. We developed the expert system using the CLIPS package from NASA. Placing the expert system at the top level provided a controller to make large-scale decisions about how to solve the entire control problem, without considering detailed control issues. Details concerning how to solve subproblems were handled by lower level control modules. With direct access to the Vsystem control database, the expert system used all pertinent information to build a model for solving the system and to reason about specific components and more general tuning issues. This top-down approach reflects a true expert's knowledge in a large system, and it provides a good framework for building proper knowledge representations.

With the expert system in control, an object representation of the accelerator was necessary for reasoning about the beamline and for easy manipulation of structured objects reflecting the modularity of the beamline. Creating an object representation of the system within CLIPS enabled us to place knowledge about specific components within the component representation while maintaining a knowledge base representing facts and rules about the entire system. An object reasoning model allows appropriate encapsulation of knowledge with system objects, modularity of reasoning, and the possibility of distributed control. It is feasible with this system to make top level decisions and then provide control information to a distributed set of semi-autonomous control agents (objects).

Top-level control by an intelligent reasoning system facilitates breaking both problem and solution spaces down into well defined, easy to reason with subcomponents. We began by separating beamline components into groups by both functionality and control characteristics. By looking at the characteristics of each component, we can develop multiple partitions that imply certain types of solutions. Once the solution space has been well partitioned, an appropriate set of solutions is defined to operate on those partitions. The top level reasoning system can focus a particular partition and determine the best solution strategy for its resolution.

NEURAL NETWORKS

A neural network is a group of individual processing elements, often divided into layers. The neural network passes the results of computations between layers and finally to an output layer. The individual processing elements, roughly analogous to biological neurons, combine inputs from multiple input paths and create an output using a transfer function. Neural processing elements can be combined into a variety of

architectures and, along with associated training functions, can learn and recall non-linear functions and patterns. Neural networks have the additional benefits of being able to function in the presence of incomplete or noisy input data, and processing inputs in parallel.

We chose a three-layer backpropagation network because of the simplicity of the task (learning a linear relationship in the presence of noise) and the straightforward representation. We were able to directly map network input and output nodes to BPMs and SMs respectively. The network was then trained to recognize causal relationships between changes in BPM readings and magnet adjustments. A fully trained network was given desired BPM changes as inputs. The network produced magnet adjustments that would cause the changes.

Because we wanted the neural network to learn relationships on the running beamline and to adjust its weights accordingly, we attempted to train the network on a run-time data set. We accomplished this task by allowing the system to make random adjustments to magnets on the beamline model and record the resulting BPM changes. This design produced a real-time training instance for the network. We then fed the BPM changes through the network to generate a set of predicted magnet adjustments. The system calculated the difference between predicted and actual magnet adjustments and backpropagated the error through the system.

While the neural network was able to learn SM/BPM relationships using limited training cycles, it was not able to converge on more complex problems. The neural network failed for two important reasons. The first reason was inadequate training data. Because the network attempted to gather real-time data about the current state of the system, it could not produce training data faster than it could evaluate SM/BPM changes in the system. While the network could typically improve its performance by continuing to take samples of the system and produce training instances, it would take too much on-line beam time to produce enough training data to adequately train the network.

Even if a training set were available, the network suffered from a greater problem. When the network generates training data as SM/BPM pairs and attempts to learn a causal relationship between random SM changes and resulting BPM changes, the network is not directed toward specific solution methods for adjusting BPMs. Potentially, a large number of SM adjustments could produce the same effect on BPM readings, given SMs in the same axis using separate power supplies.

ADDITIONAL CONTROL METHODS

We also investigated several other control methods for these problems. These included an analytic control algorithm, fuzzy logic control and genetic algorithm searches.

The analytic technique for steering control relies on beamline behavior consistent with a simple linear model. After the expert system determines a set of components that make up a steering section, it measures the derivative between steering

magnet power source currents and beam monitor readings. The controller adjusted power to the SMs in the section and recorded resulting changes in BPM readings. The expert system then built an appropriate system of equations and solved them using gaussian elimination (reduction). The analytic method makes no attempt to filter noise or eliminate component errors. In general, the analytic method provides an accurate solution given large signal-to-noise ratio and properly functioning beamline components.

We cannot expect a purely analytic solution to adequately tune a beamline in most cases, especially during initial startup. One of the conditions that causes difficulty for tuning is beam fluctuation or jitter. Fuzzy logic is used in the beamline controller for reasoning about real-valued beamline data in the presence of noise or in situations where analytic methods have failed. The fuzzy logic steering solution used fuzzy rules about BPM relationships to follow a hill-climbing algorithm toward a good solution. Not only do fuzzy rules allow expert systems to reason about real-valued data without relying on specific values for rule boundaries, but they also allow reasoning about how data will be measured and evaluated. The expert system is able to modify the meaning of a set membership depending upon the specific problem being solved, the accuracy required, and the state of the system.

The fuzzy logic solution did a good job of quickly moving to an approximately correct solution, but tended to oscillate around a very accurate tune. We noticed that the accuracy of the fuzzy solution depended greatly on the quality of the knowledge we placed in the system. For example, a pure hill-climbing fuzzy system that only attempted to minimize BPM error tended to find local minima. When the rules were modified to evaluate BPM ratios and isolate specific magnets for adjustment, the fuzzy solution tended to find better solutions fairly quickly.

The genetic algorithm (GA) is an appropriate heuristic for focusing control because it can search large solution spaces in non-linear domains. The GA for steering control used genetic operators that modified magnet strengths according to a preset probability distribution. We implemented the focusing algorithm using fuzzy genetic operators which modified magnet strengths according to a fuzzy pattern. Fuzzy patterns eliminate the need for a priori determination of magnet adjustment strengths and patterns. The focusing GA was built with the realization that wire scanners cannot deliver real-time continuous feedback. Trial solutions for the algorithm are evaluated by actual testing on the simulated beamline. Since typical solution patterns can be determined for focusing, we used a special genetic operator to search the solution population for unwanted solution patterns (as determined by the expert system) and replace them with patterns representing good possible solutions. Fuzzy pattern matching and replacement guides the GA toward certain solutions and away from others, according to knowledge about typical solutions in the expert system. The algorithm can still perform a global search over the solution space, and it can still converge on a solution with a pattern differing from suggested "good"

patterns. The guided solution was able to focus the periodic line in under 100 trials and to greater than expected accuracy.

PROJECT STATUS

The first phase of this project is nearing completion. An application has been submitted for additional funding. Work is continuing to develop the next generation of the system based on the results of the first prototype. One of the major goals of the second phase of this project will be to test the controller at several operational accelerator facilities to determine successful it is under realistic conditions. The additional tests for the second phase of the project will be to expand the complexity of the simulated tuning problems to further stress the capabilities of the controller.

Using knowledge gained during the first phase of this project, we have prepared a clear plan for developing a full, intelligent control system for accelerators. In the most effective system, an expert system coordinates the activities of a set of independent processes controlling small subsystems of the accelerator. The expert system manages the overall tuning process by identifying and configuring subgoals based on the overall goal for the accelerator. These subgoals are then either subdivided further or are assigned to a suitable method, based on the goal and the operational situation. Provided with a varied set of methods, an expert system can overcome limitations in any particular control method by substituting a solution method well matched to the goal for a particular subsystem.

ACKNOWLEDGMENT

REFERENCES

[1] S. Clearwater and W. Cleland, " A Real-Time Expert System for Trigger Logic-Logic Monitoring ," *Proceedings of the International Conference on Accelerator and Large Experimental Physics Control Systems*, Vancouver, B.C. (November 1989)
[2] D. Nguyen, and M. Lee, "Accelerator and Feedback Control Simulation Using Neural Networks," (SLAC-PUB-5503, May 1991)
[3] D. Schultz, "The Development of an Expert System to tune a Beam Line," *Proceedings of the International Conference on Accelerator and Large Experimental Physics Control Systems*, Vancouver, B.C. (November 1989)
[4] K. Brown et al. "TRANSPORT A Computer Program for Designing Charged Particle Beam Transport Systems," (SLAC-PUB-91 Rev 2, May 1977)