

An Architecture for Intelligent Control of Particle Accelerators

William B. Klein, Robert T. Westervelt
Vista Control Systems Inc., Los Alamos, New Mexico 87544
and
George F. Luger
University of New Mexico, Albuquerque, New Mexico 87131

ABSTRACT

In this paper, we discuss results of combining various methodologies from the field of artificial intelligence into the design of a control system for accelerator tuning. Our architecture brings together state space search and rule-based reasoning with adaptive/learning algorithms such as fuzzy logic, neural networks and genetic algorithms. We discuss current efforts extending the system to include a general purpose hierarchical control paradigm, parallel distributed reasoning, an object-oriented reasoning structure and additional heuristic control methods.

1.0 PROBLEM OVERVIEW

The goal of this project is to develop a flexible intelligent controller that can reduce the tuning time and the need for human intervention in the control of a particle accelerator. We also wish to produce better and more stable tunes than those that are now achieved by human operators. Various approaches have been taken to automate accelerator control [2], [8], [12], with varied degrees of success. Most effort has been directed toward solving specific problems for a particular facility and little effort has been directed toward developing more general solutions applicable to the diverse specifications and tasks of a number of different accelerators. This paper reports the status of our continuing research including efforts toward building a general purpose control system.

To build an environment for testing our control algorithms we interfaced TRANSPORT [1], a standard accelerator modeling program, to Vsystem [3]. Vsystem is a commercial software product for developing control systems. Vsystem provides a distributed database and tools for accessing real-time data, as well as a graphical environment for display and control of database channels. We began by developing a computer model to simulate steering and focusing elements used in beam transport. Noise and error effects including initial beam jitter, electronic offset and drift and random gaussian noise were added to data signals to approximate actual tuning conditions. Time dependent device behavior was also included in the simulation. Figure 1 depicts the simulated beamline.

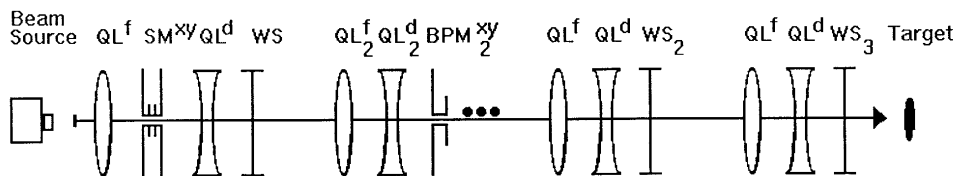


Figure 1. Typical Beam Transport Line. Beam behavior is simulated using transport modeling code. Measurement and control occurs through Vsystem control channels.

2.0 CONTROL DESIGN ISSUES

We considered many control design issues during the development of the control system. These include: 1) adaptive vs. non-adaptive control, 2) optimal vs. "good enough" solutions, 3) scalability, 4) determination of failure conditions, 5) online and offline learning, 6) stability in a heuristic control environment. Our

design includes a hierarchical decision maker which determines appropriate control techniques according to the issues listed above. The controller is able to choose from a variety of control techniques and substitute different methods according to the state of the system.

Our design uses an expert system at the top level for reasoning and control. An expert system is a computer program that uses an explicit knowledge base, often directly taken from human experts, along with logical reasoning to solve complex, real-world problems [7]. Placing the expert system at the highest level provides a controller capable of making decisions about the control problem in a global context, without considering detailed issues; context specific subproblems are handled by lower-level control modules. With direct access to the Vsystem control database, the expert system applies all pertinent information to build a model for solving the system and to reason about specific components and more general tuning issues.

We used CLIPS, a forward chaining object-oriented expert system shell [4] for building knowledge structures that represent expert knowledge in both the problem and solution domains. We separated the beamline components into groups by function and control characteristics and then developed partitions that implied certain types of solutions. That is, we developed control structures which included facts and rules about beamline interactions as well as control techniques for solving classes of problems.

We constructed CLIPS objects for beamline components which represented both physical entities and control characteristics. These objects included static information about beamline placement and orientation, as well as methods for data collection and control during operation. Creating an object representation of the system within CLIPS enabled us to place knowledge about a specific component within its representation while maintaining a separate knowledge base representing facts and rules describing the entire system. An object reasoning model allows appropriate encapsulation of knowledge with system objects, modularity of reasoning and the possibility of distributed control.

3.0 A CONTROL METHODOLOGY

In this section we outline a number of heuristic methods for control of the partitioned submodules of the accelerator. These control methods include neural (or connectionist) networks, fuzzy logic and genetic algorithms, as well as more traditional analytic methods. In using these heuristic methods we make certain basic assumptions about the control problem based on suggestions from Ross [9]:

- 1) Beamline behavior is observable and controllable. The control techniques used here rely on measurable state input and output variables. Human agents, through constant monitoring can control the system.
- 2) There exists a method for encapsulating knowledge about beamline control within the heuristic methods. This may come from neural network learning algorithms, a priori rule-based knowledge, or inherent knowledge encoded in the genetic algorithm population.
- 3) One or more solutions exist. The set of control variables is sufficient to produce correct beamline behavior.
- 4) A "good enough" solution is acceptable. We will identify a small error range within which all solutions are valid.
- 5) Optimality and stability may be shown through data flow analysis and empirical methods, rather than through formal proofs. Because many of our control heuristics use inexact methods, formal proofs are inappropriate if not impossible.

Keeping these assumptions in mind, we adapted the following techniques for use in accelerator control:

We experimented with multilayer perceptron networks to attempt to learn the relationships between control and feedback components in the beamline. Unlike the traditional use of neural networks in control [8] we did not model beamline behavior, but instead used the network to discover causal relationships between beamline components. Although preliminary efforts were unsuccessful, work is continuing using connectionist systems for direct beamline control [12].

Analytic techniques for control rely on beamline behavior consistent with a simple linear model. A straightforward analytic method makes no attempt to filter noise or eliminate component errors. In general, this 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. Fuzzy logic is used in the beamline controller for reasoning about real-valued data in the presence of noise and in situations where crisp analytic methods have failed. Fuzzy logic attempts to categorize real data sets with ambiguous boundaries. We may consider the data we receive from beamline measuring devices as ambiguous or imprecise, because data measurement involves errors of unknown distribution and sometimes from unknown sources. We can capture a human operator's reasoning about this imprecise data by specifying linguistic variables comparable to the fuzzy sets which match the operator's (implicit) fuzzy categories. We implemented fuzzy logic versions of analytic solutions successfully for noisy steering simulations [12].

Not only do fuzzy rules allow expert systems to reason about real-valued data without crisp data boundaries, they also allow reasoning about how data will be measured and evaluated. An expert system could refine the meaning of a fuzzy control variable during different stages of the solution. This refinement relates to the context dependent nature of fuzzy membership functions and the ability to reuse fuzzy rules in both coarse and fine grain solutions. For example, a rule governing steering behavior may state that a *small* adjustment should be made when the error is *small*. The term *small* may have a different meaning for error and adjustment. Furthermore, as subsequent adjustments continue to decrease the error, we may need to adjust membership functions representing *small* to ensure convergence. The expert system can change the membership function associated with a variable depending on the specific problem being solved, the accuracy required and the current state of the system.

The genetic algorithm offers an appropriate heuristic for focusing control because it can search large solution spaces in non-linear domains. The genetic algorithm is particularly useful when the controller must function using incomplete information or when the system behaves abnormally due to component failure or other unpredictable situations. We implemented a genetic algorithm for beam focusing using genetic operators which modified magnet strengths according to fuzzy patterns. Fuzzy patterns eliminate the need for a priori determination of magnet adjustment strengths. 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 solutions fitting good patterns. We found that the fuzzy pattern matching solution focused the simulated periodic line in fewer than 100 trials and to a greater than expected accuracy [12].

4.0 A SYNTHESIS OF CONTROL MODULES

By incorporating the solution methods of Section 3 into a single system, we are developing a powerful integrated problem solver that addresses many beamline tuning problems. Modifying existing solution algorithms and adding new solution strategies can enhance the quality and speed of the system within the current framework and generalize our solutions for use on other accelerators.

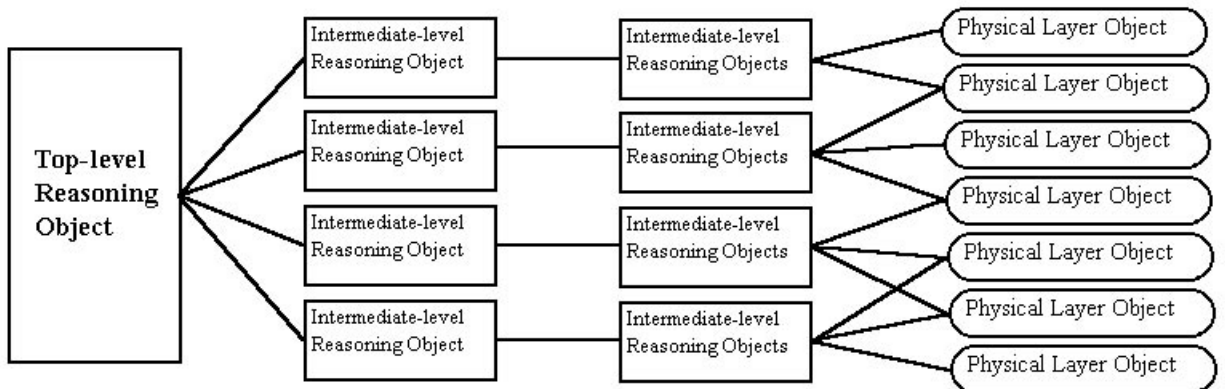


Figure 2. A typical object reasoning hierarchy.

By placing the expert system as the highest-level decision maker for the controller, we use expert knowledge to break the control problem into solvable units and then determine an appropriate solution strategy for various beamline problems. Keeping decision making isolated at a single level, however, causes problems with rule complexity and problem decomposition. Our current system makes use of a structure which represents a control hierarchy at higher levels integrated with a physical component hierarchy at lower levels. Figure 2 illustrates a multilevel reasoning hierarchy.

This modular decomposition of complex problems into multiple interacting subcomponents is central to our approach. The object oriented methodology provides data structures that "wrap" the submodules in a module hierarchy, where each module contains knowledge (coded as facts, rules, and procedures) describing its functionality, as well as sets of methods for cooperating with other modules. Together the interacting modules make up the larger system. This approach to problem solving is called by the artificial intelligence research community a *solution strategy based on the interactions of autonomous intelligent agents* and has been used in a number of different application areas including electricity transport management [6] and building environment control [5].

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 tuning process by identifying and configuring subgoals based on an overall goal for the accelerator. These subgoals are then either subdivided further or assigned a suitable solution strategy based on the goal and the current operational state. We have found that an expert system equipped with a "toolbox" of control methods can overcome limitations of any one control method by substituting a specific control strategy based on a particular subsystem goal.

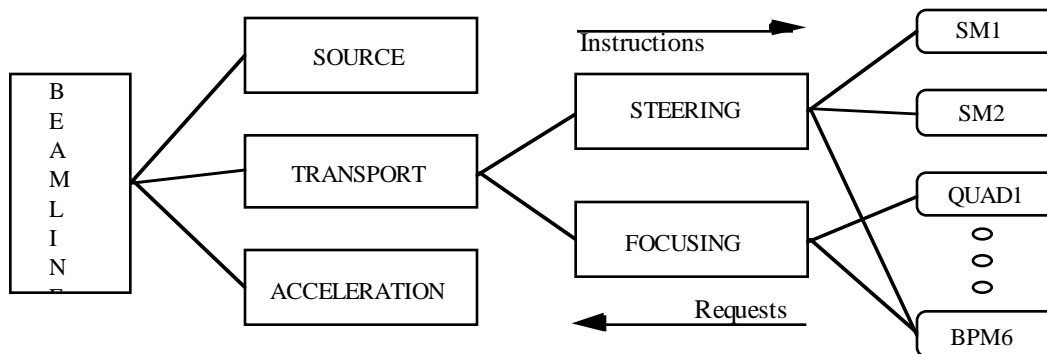


Figure 3. A hierarchy for a simple beamline with transport.

Figure 3 illustrates an example control hierarchy for a simple beamline consisting of a beam source, a transport section, and an accelerator. A top level control object representing the beamline contains high level information about each of its subcomponents (source, transport, acceleration) and their interactions. The beamline object also keeps track of global events and coordinates activity for responding to alarms or errors. The transport object contains knowledge about the transport section and information about interactions of its subcomponents, steering and focusing. Steering and focusing objects contain specific information regarding beamline components and problem solving methods. At each level, an object's parent serves as an intermediary for communication between control components. For instance, the transport object can send a message to the beamline object requesting that the input beam be more accurately centered. The beamline object may then send instructions to the beam source object to request the change or deny the request and send new instructions back to the transport object. Such communication and object interaction may occur at any level in the hierarchy.

5.0 SUMMARY AND CONCLUSION

The concept of object models for accelerator systems is a methodology gaining a large following in the accelerator control community. Work has been done at numerous sites to develop an object framework for describing accelerator control applications [11].

An expert system with general knowledge of the control domain exists at the top level for coordination and control of beamline subcontrollers; smaller domain-specific rule sets exist throughout the object oriented component module hierarchy. Distributing knowledge throughout the system has a number of advantages: 1) Rule sets are typically smaller; large rule sets indicate the need for breaking down the problem into smaller components, 2) Knowledge resides at the appropriate level in the system, so control objects can make domain-specific decisions without relying on a higher level control object, 3) Reasoning is faster; the conflict set for any one rule base is smaller and independent activations may be fired simultaneously in the distributed environment.

Further discussion of our research may be found in [12], [13].

6.0 ACKNOWLEDGEMENTS

This work was supported by a grant from the Department of Energy under the SBIR program (Grant #DE FG05-94ER81897). We would like to thank Andy Jason of AOT-1 at Los Alamos National Laboratory for his invaluable assistance in developing control scenarios and for his careful explanations of the underlying physics concepts involved.

7.0 REFERENCES

- [1] Brown, K., Rothacker, F., Carey, D. C., and Iselin, Ch. (1977). "TRANSPORT: A Computer Program for Designing Charge Particle Beam Transport Systems," SLAC-PUB-91 Rev 2.
- [2] Clearwater, S. and W. Cleland (1989). "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.
- [3] Clout, P. (1993). "The status of Vsystem," *Proceedings of the Third International Conference on Accelerator and Large Experimental Physics Control Systems*, Germany.
- [4] Giarratano, Joseph and Riley, Gary (1989). *Expert Systems: Principles and Programming*, Ch. 7, Boston, MA, PWS-KENT Publishing Company.
- [5] Huberman, Bernardo and Clearwater, Scott H. (1995). "A Multi-Agent System for Controlling Building Environments," *Proceedings of the First International Conference on Multi-Agent Systems*, San Francisco, CA., MIT Press.
- [6] Jennings, N. R., Corera, J. M., and Laresgoiti, I. (1995). "Developing Industrial Multi-Agent Systems," *Proceedings of the First International Conference on Multi-Agent Systems*, San Francisco, CA., MIT Press.
- [7] Luger, G.F. & W.A. Stubblefield (1993). *Artificial Intelligence: Structures and Strategies for Complex Problem Solving*, Reading, Ma: Addison Wesley.
- [8] Nguyen, D., M. Lee, R. Sass, and H. Shoaee (1991). "Accelerator and Feedback Control Simulation Using Neural Networks," SLAC-PUB-5503.
- [9] Ross, T. (1995). *Fuzzy Logic with Engineering Applications*, New York, NY: McGraw-Hill, Inc.
- [10] Schultz, D. (1977). "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.
- [11] Skelly, J.F. (1991). "Object-Oriented Programming Techniques for the AGS Booster," *Proceedings of the International Conference on Accelerator and Large Experimental Physics Control Systems*, Tsukuba, Japan.
- [12] Westervelt, R.T., Klein, W. B., and Luger, G. F. (1995). "Framework for a General Purpose Control System for Particle Accelerators.," *Proceedings of the IEEE Particle Accelerator Conference*, Dallas, Tx.