An Intelligent Control Architecture for Accelerator Beamline Tuning

William B. Klein, Carl R. Stern
Vista Control Systems, Inc.
134B Eastgate Drive, Los Alamos, NM 87544
Voice: (505) 277-9140, Fax: (505) 277-6927
klein@vistanm.com, stern@vistanm.com

George F. Luger, Eric T. Olsson
Department of Computer Science
University of New Mexico, Albuquerque, NM 87131
Voice: (505) 277-3204, Fax: (505) 277-6927
luger@cs.unm.edu, eolsson@cs.unm.edu

Abstract

This paper discusses a new architecture for accelerator tuning that combines heuristic and knowledge based methods with traditional approaches to control. Control of particle accelerators requires a hybrid architecture, which includes methodologies for planning, intelligent search, and pattern recognition. Control is distributed and hierarchical to utilize parallel problem-solving in the face of time-sensitive control requirements and to decompose complex control problems into more manageable subtasks. For perspective, we discuss past attempts at accelerator control and why these attempts left many issues unresolved. We describe the details of our control architecture along with its motivation. We then report the results of deploying and testing it at two accelerator facilities. This paper ends with a discussion of the commercial importance of this work.

The Accelerator Control Problem

Tuning particle accelerators is time consuming and expensive, with a number of inherently non-linear interactions between components of the system. Conventional control methods have not been successful in this domain, and the result is constant and expensive monitoring of the systems by human operators. In recent years with isolated successes, advanced technologies such as expert systems, neural networks, and genetic algorithms have been applied to the individual pieces of this problem.

There are many different tasks involved in control of a particle accelerator facility. We initially focused our efforts on tuning an accelerator beamline, which consists of a number of elements designed to either effect the beam using fields or to monitor the beam in a variety of other ways. Figure 1 shows a typical accelerator beamline which includes trim magnets for steering, quadrupole magnets for focusing, Faraday cups and stripline detectors for measuring current, and profile monitors for measuring beam size and position. Various components are placed along the beamline by design to produce specific effects in a known way. Unfortunately, real systems rarely work as they are designed. Problems arise from imperfect beam production, remnant magnetic fields, poorly modeled beam behavior, misplaced or flawed control elements, and changes to the design or use of the facility after it has been built. Beamline designers consider these problems and build diagnostic components into the beamlines. Profile monitors and current detectors are used to measure beam parameters throughout the line to provide information for verifying or correcting beam characteristics. Even so, imperfect detectors, system errors, and noise due to various effects cause beamline control to be difficult at best.

Given these challenges, it is hard to imagine any system capable of tuning a beamline to an acceptable measure. Expert physicists, however, accomplish this task every day. They do this by using a variety of tools for measuring and learning the current beamline behavior, including adjusting control elements to modify the beam, and then testing their results and re-planning their actions. Through extensive knowledge engineering we have observed a number of important characteristics of the accelerator control

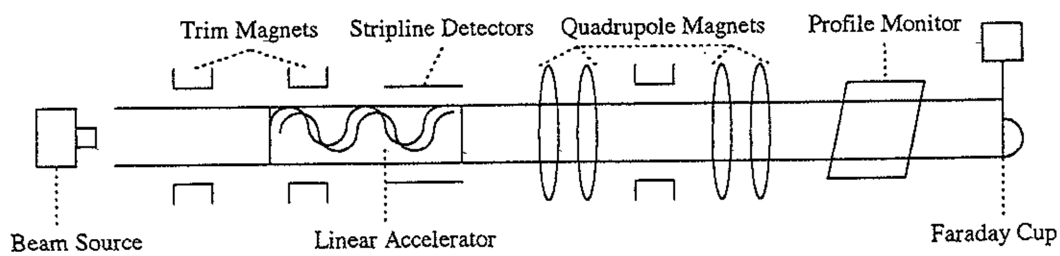

Figure 1. A typical accelerator beamline.

Klein, W.B., Stern, C., Luger, G., and Olsson, E. "An intelligent control architecture for accelerator beamline tuning." *In Proceedings of Innovative Applications of Artificial Intelligence*. Cambridge: MIT Press (1997).

process:

- Beam tuners combine analytic and theoretical knowledge with heuristic search and practical experience to produce good tunes.
- Humans know when to trust system data, and when to ignore noisy or incorrect readings.
- Expert beam tuners use many of the same search algorithms applied in traditional AI systems.

Combined, these characteristics describe many of the attributes of an automated system for control, and suggest that such a system could perform as well or better than a human at the beamline tuning task. Control systems have indeed been built to perform some of the tasks which human operators perform in beamline tuning. A brief list of some of those attempts follows.

Previous Attempts at Accelerator Control

An example of conventional accelerator control is a project by Himel et al. (1993) which used analytic methods for noise canceling at the Stanford Linear Collider. This project applied MIMO (Multiple Input Multiple Output) adaptive noise cancellers to seven beam-steering feedback loops operating on the same beam. This effort was successful at providing a supervisory filtering mechanism for a set of parallel tuning controllers.

Neural networks have been applied to accelerator control for actual manipulation of control parameters as well as for simulation. Nguyen et al. (1991) applied single layer neural networks for simulating effects of steering magnets over a series of 16 beam position monitors. Howell et al. (1990) used neural networks for modeling and control of a negative-ion accelerator source, but with limited success.

The SETUP program developed at CERN (Bouche 1995) is representative of efforts to apply AI techniques to small subsystems. SETUP is only used for pre-control equipment setup. The program uses an object-oriented description language for representing control actions. The reasoning system searches the oriented graph defined by an object description to make decisions about equipment setup without human assistance. The program provides a good example of using object models for control decision making. It does not attempt to perform real time control or use on-line feedback from the system.

The ZEUS project at Deutsches Elektronen Synchrotron, Germany (Behrens et al. 1996) is an effort toward automating a substantial portion of an accelerator facility. The ZEUS expert system, ZEX, is a blackboard-based architecture designed to add human experience for supervisory control. ZEUS and ZEX work together to provide slow control, data acquisition, data quality monitoring, and run control. The reasoning architecture is a forward chaining production system manipulating a complex hierarchy of control objects. Data is gathered and symbolized using syntactic pattern recognition at lower levels by clustering observed phenomena. Knowledge sources in the blackboard attempt to recognize control state over posted patterns and contribute to a global control solution. While this project constitutes a significant achievement in accelerator control automation, it is a very special purpose project requiring its own crew of experts.

The ABLE and GOLD systems developed by Clearwater and Lee (Lee 1987) were prototypes for real-time control of an entire beamline. ABLE used simple rule-based reasoning to perform tests and directly manipulate a beamline simulation to correct beam transport errors. ABLE was successful on a number of simple simulations, but has never been tested as a general solution or on a real accelerator.

Other attempts at intelligent control for accelerators include the ISIS tune advisor (Schultz et al. 1990), the LAMPF Beam Loss Expert (Clearwater et al. 1986), and a learning system based on RL4 (Clearwater et al. 1990). The ISIS tune advisor and LAMPF Beam Loss Expert were both expert systems for indirect control (advising human operators) which were never implemented as general or real time control solutions. The learning system used knowledge-based induction for off-line learning of beam position monitor placement, but was not implemented as a general learning algorithm.

In summary, many attempts based on conventional control algorithms have been made to automate part or all of the accelerator control process. Other attempts that have included heuristic or "intelligent" approaches to the control problem have selected a small subset of control technologies, for instance model-based control or supervisory control using expert systems. This piecemeal approach to applying AI has been valuable for determining the usefulness of a number of approaches to accelerator control, but is not satisfactory as a solution for total automation of the process. We propose a technique which combines different methodologies and builds upon their strengths (Klein et al. 1997).

A New Approach

We have identified two different sources of control information which we believe must *both* be incorporated into any successful automated control system. The first source includes analytic domain knowledge necessary for modeling the accelerator and beamline. The second, equally important, source is experiential knowledge about the specific facility and group of components being controlled. We have found that true "experts" at beam tuning are accelerator physicists with strong theoretical background who are experienced at using modeling tools and who spend a great deal of hands on time tuning the accelerator and beamline.

Our system is based upon a distributed hierarchical architecture designed to incorporate a wide variety of representations, both analytic and knowledge-based, into a single control framework. At the heart of the architecture is a group of knowledge-based *controllers*. These controllers are hierarchically organized in a structural/functional hybrid design (see Acar et al. 1993). Controllers are responsible for making decisions about what control actions will be performed, when they will occur, and how their performance will be measured. Controllers are also responsible for reasoning about system state, diagnosing errors in control solutions, decomposing goals into tasks and actions, and initiating any necessary human interaction.

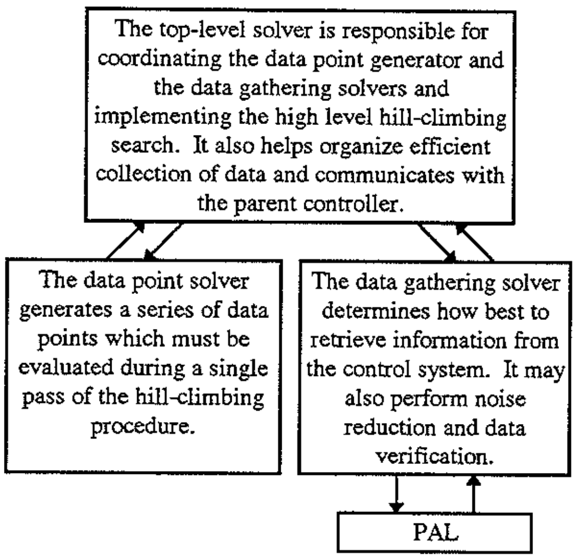

Figure 2. A hill-climbing algorithm using three solvers.

Controllers carry out plans which accomplish user-defined goals by applying various forms of domain knowledge. *Solvers* are reusable components which can be configured by controllers to apply low-level, well defined algorithms to the control process. Solvers encode procedures that can be assembled (again in a hierarchical manner) for run-time construction of control algorithms. Figure 2 shows a typical solver-based procedure for applying a search algorithm, in this case simple hill-climbing. The procedure is broken into three parts, a solver for generating data points which must be measured during search, a solver for measuring the data points using appropriate elements in the domain, and a parent solver which coordinates actions of the two children in a way that performs hill-climbing. A different algorithm, Newton's method for example, can be constructed by merely substituting the parent controller for one that applies a different top-level procedure. Different controllers may also be substituted in cases where specific constraints (e.g., noise handling, speed, etc.) are important.

Because we use a symbolic system for reasoning about the control system, raw data is rarely appropriate for direct manipulation by controllers. The same is true in reverse; a low-level interface for manipulation of control elements is usually inappropriate. For this reason we have developed an object-oriented Physical Access Layer (PAL) as an abstraction mechanism between controllers and the underlying control system. This provides a number of important advantages:

- The PAL provides a mechanism for hiding unimportant implementation details about hardware and provides a uniform interface for control access.
- Resource conflicts can be initially handled at a low level and, once identified, mediated at the controller level.
- Controllers can pass filtering instructions to the PAL to allow pre-processing of data into a representation expected by a controller. This can happen, for example, by giving the PAL fuzzy sets for classifying data, or passing a neural network encoding to the PAL.
- The system is highly portable. By abstracting underlying control elements, control algorithms can be written in a generic manner. The same set of controllers can be used at multiple accelerator facilities by exchanging the PAL.

The PAL is composed of a number of Physical Layer Objects (PLOs) which are representations of a control or diagnostic element or collection of elements. These objects can be as simple as single magnets, or as complex as non-linear tuning knobs which manipulate a series of magnets. PLOs communicate with hardware indirectly through Vsystem, a high speed software data bus (Clout 1993). PLOs can be organized in a hierarchical fashion and operate in parallel, much like controllers. The PAL provides PLOs access to a library of tools for representing and filtering data, algorithms for noise handling, pattern recognition, and feature extraction. Figure 3 illustrates the design of the PAL.

The PAL does more than provide a high-level interface to the underlying software control system. The PAL also performs low-level control over groups of components which together represent a control or measurement element. For instance, at Brookhaven National Laboratory's Accelerator Test Facility (ATF), beam measurements are usually taken through profile monitors which consist of phosphor screens that emit light when struck by electrons. The light is recorded by video cameras and the images from those cameras is recorded by a video frame grabber. The PAL hides the process of capturing beam characteristics from these devices and only exposes important features of the process, like conflicts in use of the frame grabber, or position information from the monitors.

The organization of controllers and solvers reflects an "all data is local" design. For this reason, information is only shared between controllers through globally accessible mechanisms, for instance a system model or the PAL, or through message passing between controllers. Message passing is the primary means for organizing control actions and distributing data throughout the system. Messaging typically occurs between parent and child controllers and is used to pass task information, convey system state, inform a parent of progress toward accomplishing some goal, or request assistance in satisfying a set of constraints.

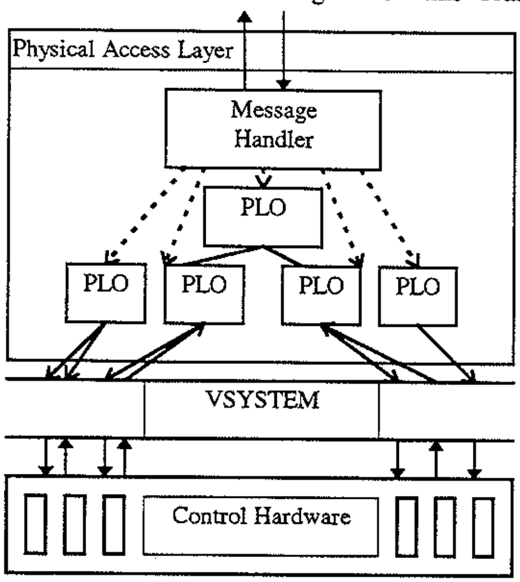

Figure 3. The Physical Access Layer is composed of abstractions called Physical Layer Objects (PLOs).

An Example

One important beamline tuning task is steering the beam through a sequence of quadrupole lenses such that any subsequent focusing of the quadrupoles does not further steer the beam. In general, this is accomplished by steering the beam through the center of the beamline. If the quadrupoles are all aligned with respect to the beamline, this produces the desired result. If the quadrupoles are not centered on the beam pipe, zero steering can still be accomplished by determining the true center of the quadrupoles and steering through it. Unfortunately, if the quadrupole lenses are misaligned with respect to each other, a perfect solution is not possible. The goal then is to steer such that focusing the quads produces a minimum steering effect.

If a perfect model of the system is unavailable, the control system must perform a sequence of actions to produce minimum quad steering. They are:

- Set all quad strengths to zero.
- Use upstream magnets to steer the beam to the center of the beam pipe as measured on two downstream monitors. Measure derivatives of the change in steering magnet strength versus position on the monitors.
- Turn on each quad, one at a time, and determine the steering effect. Re-steer the beam using previously measured steering derivatives until no quad steering occurs. Calculate the offset of the quad, reset it to zero strength, and continue.
- Use a least-squares fit of the quad offsets to determine the minimum focusing effect steer. If all quads are misaligned by the same amount, this will produce zero quad steering.
- Use an optimization algorithm to fine tune the results.

Figure 4 shows a control hierarchy for accomplishing these tasks. The minimal-steer controller begins by determining the correct set of components to use to accomplish minimum quad steering. It then sets all quadrupole magnet strengths to zero by sending a message to each of the PLOs representing quad magnets (QDP1-4). The minimal-steer controller then sends a task to the steering solver, telling it to steer to the center of the beam pipe. The steering solver performs this task by using two solvers as children, one to produce data points for calculating derivatives of magnet strengths versus position on monitors, and one to take measurements at each data point. The steering solver uses results from its children to correctly steer to the desired position. An important aspect of the steering solver is that it contains knowledge about how to correctly order the set of measurements taken by the steering gather solver such that the time spent inserting and retracting position monitors is minimized. The steering gather solver also applies intelligence by predicting and verifying measurements.

Once initial steering is accomplished, the minimal-steer controller passes a task to the quad-align solver telling it to determine the alignment of each quadrupole. The quad-align solver responds by using the quad-align data solver to produce quad settings and the quad-align gather solver to measure them, much like the steering solver triplet. Once an alignment measurement is taken, the minimal-steer controller again uses the steering solver, which has now learned the steering derivatives, to re-steer the beam to a position which it predicts will produce less steering. This procedure is repeated until the alignment of the quad is determined.

After the minimal-steer solver has determined the alignment of each quadrupole, it uses a least-squares solver to determine the minimal steer, and then the