AIM Learning ä Adaptive, Real-Time, Control Technologies

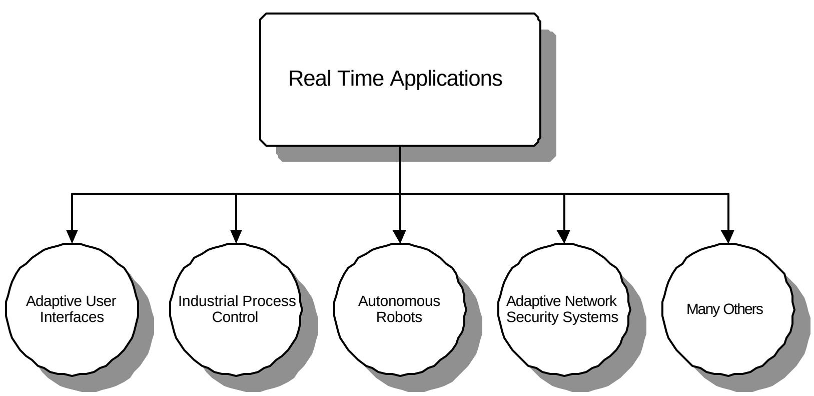
Frank D. Francone[1]
(ffrancone@aimlearning.com)

December 11, 1999

Introduction

Computerized learning techniques, such as neural networks, decision trees, case-based reasoning, genetic algorithms, and genetic programming, are methods for building predictive models. These techniques have proven to be valuable tools in a wide range of applications—business decision support, data-mining, and modeling complex physical systems, to name but a few. These traditional techniques share one common feature—a solution is learned, then that solution is applied in a static, unchanging manner to make useful predictions.

Increasingly, computerized learning is being applied to real-time control problems. Real-time control includes a broad and important class of business solutions—ranging from industrial process control, to adaptive user interfaces for computers, to control of autonomous robots.



Industrial process control would seem to be a very different problem than creating an adaptive user interface for a computer. But these types of problems share many common features:

- **Real world systems change**. A static machine learning solution does not.

[1] Frank D. Francone is the President of Register Machine Learning Technologies, Inc. He is one of the authors of one of the leading university textbooks on evolutionary computation and machine learning: *Genetic Programming, an Introduction*, by Banzhaf, Nordin, Keller and Francone (1998)

- **The real world is slow**. Feedback to the learning system is slow. Static machine learning systems are limited by the speed of the real system being controlled.
- **Preexisting knowl edge has to be relearned**. Engineers frequently know a good deal about the real-time process being controlled. But it is very difficult to incorporate that knowledge into existing approaches to control solutions.
- **Real world systems are often "mission critical."** For example, a system controlling an industrial boiler *must produce acceptable solutions every time*. One bad solution that shuts down a plant for the day is no solution.

Real-time control is, therefore, a considerably more complex problem than the traditional static modeling typical in applications such as data-mining and business decision support. This is particularly so for systems that must adapt to changes in the system being controlled. Simply put, traditional machine learning systems are not very good at acquiring knowledge and then adapting the knowledge acquired to changing circumstances.

This paper describes a two adaptive, real-time control solutions based on AIM Learning™ fast binary manipulation technology. These solutions have been in development and testing for about four years in Germany and Sweden. They combine three traditional machine learning paradigms—constant optimization, case-based learning, and genetic programming. This combination uniquely addresses most of the difficulties inherent in adaptive, real-time control systems. The system has been prototyped to control several different real time control processes with great success.

AIM Learning ä Fast Binary Learning Technology

All AIM systems have, at their core, the patented AIM™ fast binary learning technology. AIM systems operate directly on the binary machine code during the learning. This approach has is consistently between 60-200 times faster than any comparable machine learning systems. Researchers at Jet Propuls ion Laboratories have independently confirmed these figures.

AIM Learning™ fast binary manipulation is a mature, commercial technology. It is the core learning technology used in Discipulus, the first commercial Genetic Programming software. Discipulus was released in October 15, 1998. Discipulus™ is in use at Fortune 500 companies and Universities throughout the world in diverse applications such as data-mining, chip manufacture defect analysis, and embedded systems program development. Version 2.0 of Discipulus™ will be released on January 3, 2000.

The speed of this technology is a real advantage in for real-time systems. It provides learning power equivalent to sixty computers operating in parallel. This speed greatly improves the quality of solutions found by the system and makes AIM system responses fast and, effectively, synchronous.

AIM Learning ä Recurrent Programming Technology

AIM Learning Recurrent Programming Technology was developed about two years ago in Germany. It is similar in concept to recurrent neural networks but overcomes the serious problems that have plagued the training of such networks. The speed of the core AIM Technology was the key in permitting a different approach to training that is, quite simply, impossible to implement on slower, traditional systems.

This technology is designed to work on time-dependent data—such as is found in most process control applications. It has been applied very successfully to induce programs that conduct speaker independent voice recognition from raw sound samples—a very difficult time-dependent data problem that existing commercial voice recognition technologies have not yet even attempted.

A more advanced version of AIM Learning Recurrent Programming Technology combines Genetic Programming to evolve recurrent program structures and constant optimization techniques such as evolution strategies. The combined system evolves program structures that are good starting points for learning by constant optimization. Then, when the engineer wants to use an evolved recurrent program to predict a result for a particular time period, a brief constant optimization is run on the most recent data. This has the effect of customizing the evolved program structure to the peculiarities of the most current data. This approach to recurrent programming is *very* computationally intensive during training and requires either massively parallel computation or AIM fast binary learning technology.

For many real-time control applications, one or both of AIM Recurrent Programming Technologies will be a complete solution.

AIM Learning ä Real Time Control Modular System

For the most challenging real-time control applications that require ongoing adaptive learning, AIM Learning has developed a modular system that combines three different types of machine learning systems—Optimization, case-based learning, and genetic programming. This modular system leverages AIM Fast Binary Learning Technology and AIM Recurrent Programming Technology to create a system that comes online gracefully, allows engineers to embed their existing knowledge of the controlled process in the control system itself, and continues learning to adjust to the changing circumstances of the controlled process.

The System's Modular Design

The design of this system is straightforward and is diagrammed in Figure 1, *infra*. Here is a brief description of how the three modules work together.

The Control Decision Module

The Control Decision Module is solely responsible for interacting with the controlled, real-time system. All information about sensor values from the system and the performance of the system is sent to the Control Decision Module. When a control decision is required, the response module selects the best control program from the Learning Module and searches through the possible control decisions until it finds the control decision with the optimal predicted performance. Then, that control decision is sent to the real-time system for execution. After execution, the performance of that control decision is measured and the measurement is fed back into the Control Decision Module. The sensor values, the control decision, and the performance feedback is then transferred to the Memory Module.

The Memory Module

The Memory Module stores *selected* "memories" of how various control decisions actually performed in the controlled real world system. The policies used to select which memories to store are far beyond the scope of this paper. But the important point is that these policies are adjusted for a particular applications because the goals for effective control are implemented in this module.

The Memory Module is an ever changing and condensed representation of the reality presented by the real-time system that is being controlled. In a sense, this module allows AIM control systems to build a "mental model" of the reality it confronts and to constantly access that model for guidance on how to respond to new and novel circumstances.

The Learning Module

The Learning Module may include one or more different machine learning systems. The prototype models use the fast Genetic Programming engine from Discipulus. This engine uses the examples in the Memory Module to evolve programs that predict how well any particular pair of sensor values and control decisions will perform in the real world system.

In effect, the Learning Module converts the “mental model” contained in the Memory Module into a compact program that may be rapidly evaluated by the Control Decision Module to produce fast and effective control decisions. As more information about the real-time system is gathered, the “mental model” stored in the Memory Module changes and so to do the programs evolved by the Learning Module.

Advantages of the AIM Learningä Real-time Control System

The modular architecture of AIM Real Time Control Systems provides some unique advantages. They include the following:

Synchronous/Asynchronous Design

The Control Decision Module is synchronous with the real-time system being controlled. But the Memory Module serves as a buffer between the Control Decision Module and the Learning Module are asynchronous. The practical effect of this is that the high-speed Genetic Programming system in the Learning Module can operate at full speed developing control programs rather than waiting for the slow responses from the real-time system.

Incorporating Existing Engineering Knowledge of the Controlled System.

Engineers frequently have a good deal of knowledge about the operation of the real-time system being controlled. That information may be encoded into matched sensor-control decision pairs and stored in the memory module before the system ever starts operating. In fact, special policies may be placed into the Memory module giving precedence to this preexisting knowledge base.

Incorporating Existing Data Regarding Operation of the Controlled System.

Companies often have large amounts of historic data regarding the operation of the controlled system. That data may be inserted into the Memory Module and company engineers can check the performance of the programs evolved by the Learning Module before the AIM system ever comes online. In effect, the AIM modular system can be a fully functional and verifiable control system before it is allowed to control the real-time system. (Of course, the preexisting data does not allow the AIM system to devise new control strategies—that can only happen online. But the AIM system can be verifiably as effective as the existing control system before coming online.)

Fail-safe Policies.

Policies may be added to the Learning and Memory Modules that will prevent the system from generating Control Decisions in areas of the control space that are known to be mistakes. While this does not replace the need to monitor the exploratory aspects of machine learning technologies with human or a supervisory program for mission critical applications, it adds an additional layer of protection and reduces the cost of monitoring.

Controllable Exploration Policies.

Machine learning systems have to try out new solutions in order to search for and find optimal solutions. But at the same time, too much exploration or uncontrolled exploration can cause less than optimal performance in the short term, while the system is learning.

The trade-off between short term performance and exploration that best fits each real-time system is customizable. For example, very little "exploratory" behavior is desirable when a system is at equilibrium. When the system has deviated from equilibrium, however, some degree of exploration is both necessary and desirable in order to let the AIM control system learn more optimal responses. But policies may be designed to allow exploration only in relatively safe portions of the control decision space or to notify human operators when a control decision is implemented that is in a relatively unexplored area of the control decision space.

Fast and Effective Control Solutions

AIM Learning Control Systems have been applied with great success to the problem of controlling an autonomous robot in learning maze navigation. Repeatedly, robust and sensible robot behavior has emerged in minutes after commencing training. Other autonomous control systems can take weeks to learn the same behaviors.

About the Company

Register Machine Learning Technologies, Inc.™ is the company that has developed AIM Learning Technologies™. RML™ is a Nevada Corporation founded in 1996 by Frank D. Francone, Esq., Dr. Peter Nordin, and Dr. Wolfgang Banzhaf. Among them, they have published over 100 articles in peer-reviewed scientific journals and authored three full-length books on computerized automatic learning. One of those books, *Genetic Programming, an Introduction* (Morgan Kaufman, 1998) is the first university textbook in the important new computerized learning discipline of Genetic Programming.

RML successfully developed and marketed Discipulus™, the first, and clearly the fastest commercial Genetic Programming software, released in October 1998.

Much of the technology described in this paper is subject to United States patent 5,841,947 or is, in addition, patent pending.