

A mixture of intelligent and conventional control methods may be the best way to implement autonomous systems

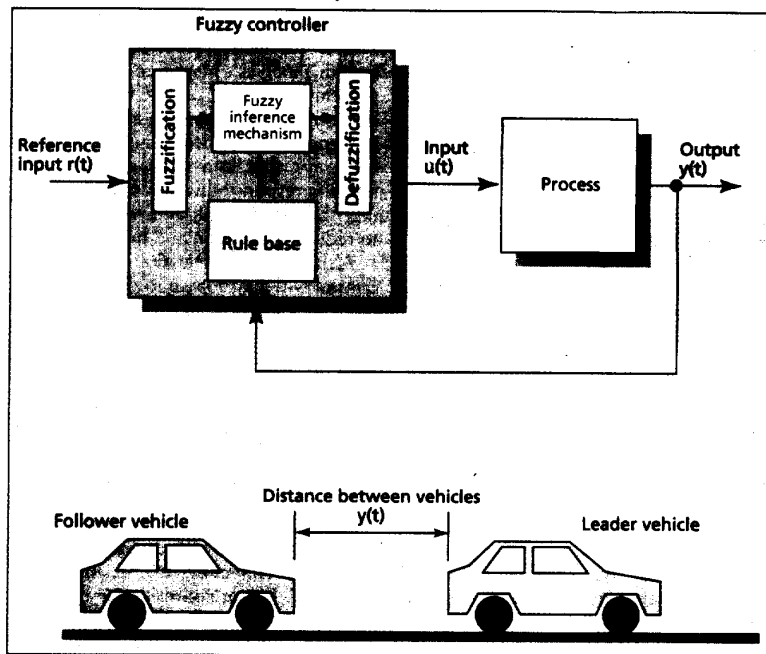
# INTELLIGENT CONTROL FOR AUTONOMOUS SYSTEMS

**I**NTELLIGENT CONTROL is the discipline in which control algorithms are developed by emulating certain characteristics of intelligent biological systems. It is quickly emerging as a technology that may open avenues for significant advances in many areas. In fact, fueled by advancements in computing technology, it has already achieved some very exciting and promising results.

Fuzzy systems, for example, despite emulating human cognition in only a simplistic manner, have dealt successfully with vibration damping in flexible-link robots and have also solved challenging problems in process control. Another type of intelligent system, the knowledge-based controller (which is based, for example, on expert or planning systems), has been employed for the management and coordination of the activities of autonomous robots. Crude circuit or computer emulations of biological neural networks have served as controllers that can learn how to control highly nonlinear systems. And genetic algorithms, based on principles of biological evolution, have been used for the computer-aided design of control systems and to automate the on-line tuning of a cargo ship autopilot control algorithm.

Unfortunately, along with these genuine achievements in intelligent control, there have also been exaggerations and inflated claims. In particular, some proponents of intelligent control systems like to say (and write) that conventional con-

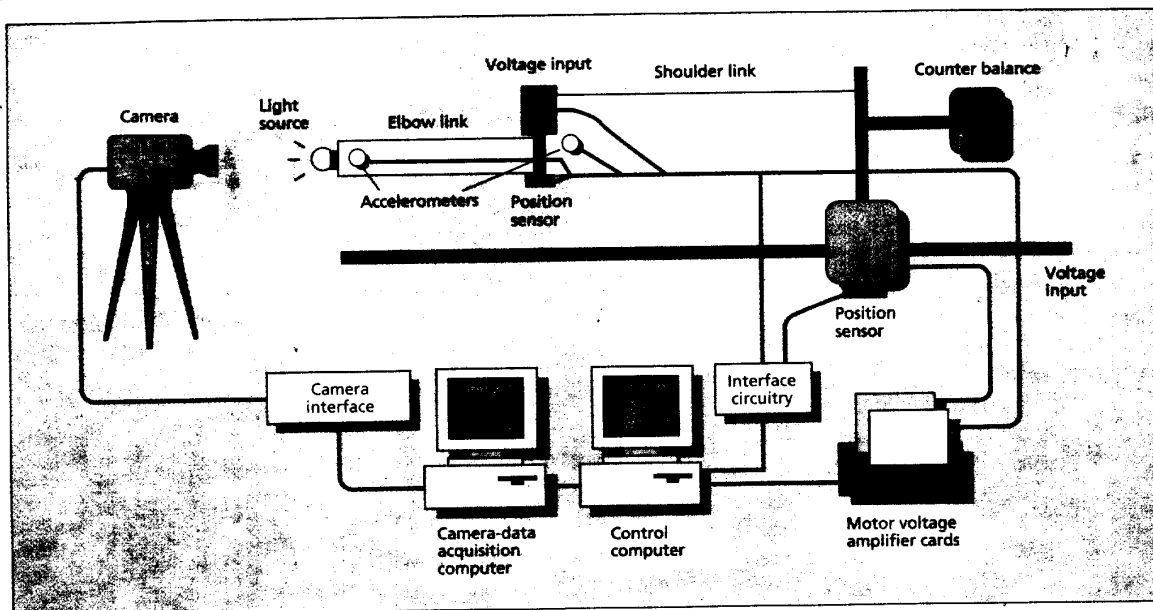
trol technologies are incapable of handling nonlinear systems and system uncertainties. The fact is that "conventional" techniques have evolved substantially over the past several decades. Proportional-integral-derivative (PID) control and state-space and frequency-domain methods, optimal control



▲ The controller in a typical fuzzy control system [top] consists of a rule base, a fuzzy inference mechanism, an input fuzzification interface, and an output defuzzifier. In the cruise-control application illustrated [bottom], the system continually adjusts the throttle,  $u(t)$ , in an attempt to minimize the difference between the desired vehicle spacing,  $r(t)$ , and the actual spacing,  $y(t)$ .

ILLUSTRATIONS: MORRIS KHAN

KEVIN M. PASSINO  
The Ohio State  
University



▲ This two-link flexible robot has independently movable shoulder and elbow joints. In the absence of feedback [see the red response curve on the next page], the end of the robot's arm will tend to vibrate quite a bit after a 90-degree slew is commanded for each link.

and robust control, the Kalman filter, adaptive control, and Lyapunov techniques, to name a few, have been highly successful in solving problems in many areas. Among the areas: vehicular control, weapon systems, robotics, manufacturing, power systems, spacecraft, aircraft control, and process control.

Another problem with intelligent control is that some engineers get so excited about the very idea of emulating intelligent behavior (whatever that means) that they tend to lose their objectivity about it. Clearly, it is necessary to ask of this technology, as for any other innovation, three important questions: for which problems, if any, can it outperform tried-and-true conventional techniques? Can its behavior be verified by modeling, simulation, nonlinear analysis, and experimentation—as is done for conventional control systems? And, will it stand up to objective cost-benefit analyses and the test of time?

Regardless of the successes of intelligent control, there is a second closely related, but more important, trend in the field of control today—the effort to integrate the functions of isolated subsystems to form highly autonomous systems that can perform complex control tasks without human help. This trend is gaining momentum as control engineers, having solved many problems, are seeking control challenges in which broader issues must be taken into consideration.

For instance, in military aviation, engineers are moving on from traditional terrain-following, terrain-avoidance control systems to a "pilot's associate" computer

program that integrates the functions of mission and tactical planning into a single system, much as a human co-pilot does. In the emerging area of intelligent vehicle highway systems, to take another example, engineers are designing vehicles and highways that can fully automate the human's responsibilities in steering, braking, throttle control, and route selection to reduce congestion and improve safety.

Although it is clear that conventional control will play a large role in the development of such highly automated systems, it is also possible that highly autonomous behavior may be more easily achieved with intelligent controls. Even more likely, a combination of the two approaches may prove to be the best solution.

To determine the best overall engineering methodology for the development and deployment of autonomous systems—especially when safety issues are of concern—it is helpful to have a framework, or architecture, for the incorporation of intelligent control techniques into autonomous systems. Before getting into that area, however, it is best to review the techniques of intelligent control and to highlight those of their characteristics that have proven to be especially useful in particular applications.

### Fuzzy control

The workings of intelligent controllers are usually described by analogies with biological systems—for example, by looking at how human beings perform control tasks or recognize patterns and make decisions. One of the most widely

publicized techniques for embodying human-like thinking into a control system is fuzzy control.

A fuzzy controller can be designed to roughly emulate the human deductive process—that is, the process people use to infer conclusions from what they know. A typical fuzzy controller consists of four main components: a rule base, a fuzzy inference mechanism, an input fuzzification interface, and an output defuzzification interface [see upper part of drawing on the previous page].

The rule base holds a set of IF-THEN rules that quantify the knowledge that human experts have amassed about solving particular problems. It acts as a resource to the fuzzy inference mechanism, which makes successive decisions about which rules are most relevant to the current situation and applies the actions indicated by those rules.

The input fuzzifier takes the "crisp" numeric inputs to the system and, as its name implies, converts them into the fuzzy form needed by the fuzzy inference mechanism. At the output, the defuzzification interface combines the conclusions reached by the fuzzy inference mechanism and converts them into a crisp numeric value as an output.

As an example of the working of a fuzzy control system, consider an advanced automotive cruise-control system [see bottom of drawing on previous page]. The design objective is to specify the rule base to represent the manner in which a human driver in the follower vehicle would act to regulate the inter-vehicle dis-

tance. The rule base contents would be a set of IF-THEN rules of this general type:

**IF**  $e(t)$  is positive-small  
**and**  $de(t)/dt$  is positive-medium,  
**THEN**  $u(t)$  is positive-medium

**IF**  $e(t)$  is positive-small  
**and**  $de(t)/dt$  is negative-medium  
**THEN**  $u(t)$  is positive-small

where  $e(t)$  is the error between the desired and the actual inter-vehicle spacing and  $u(t)$  is the throttle input to the follower vehicle.

The first rule listed above quantifies the driver's knowledge that if the inter-vehicle distance error is small—that is, the actual vehicle spacing,  $y(t)$ , is almost the same as the desired spacing,  $r(t)$ —and the error is increasing at a moderate rate, then the throttle input of the follower vehicle should be positive and large enough to counteract the growing gap.

The second rule indicates that if the error is again small, but this time it is decreasing at a medium rate, then only a small throttle input should be applied, since the follower vehicle is already moving to correct for the error in the inter-vehicle spacing.

A complete rule base consists of a whole set of such rules in which linguistic descriptions like "positive-medium" are given precise meaning by fuzzy logic. Since several standard methods exist for implementing the fuzzy inference mechanism, the main problem in designing a fuzzy control system is the specification of the rule base. Overall, the fuzzy control design methodology provides a heuristic technique for constructing nonlinear controllers, and this seems to be one of its main advantages.

**Fuzzy supervisory control**

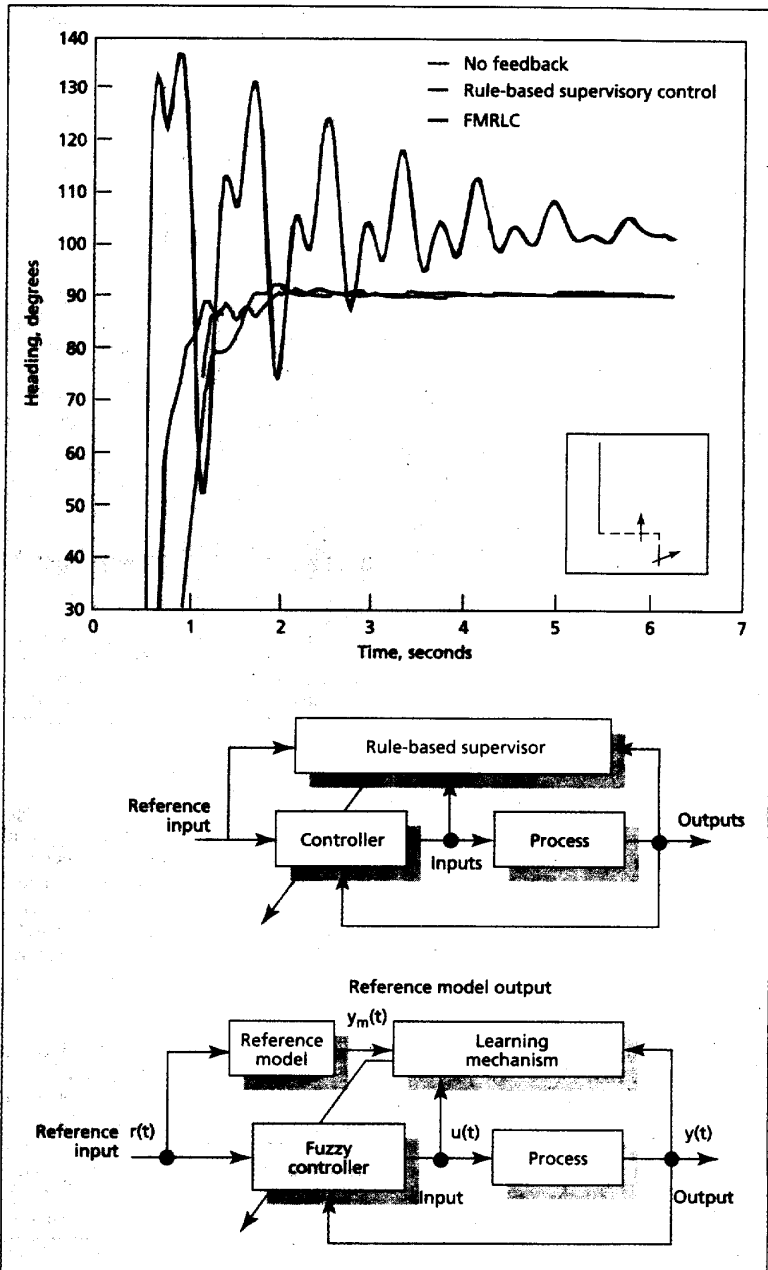
It is often the case that higher-level knowledge about how to control a process is available along with the lower-level data on which simple control systems operate. Examples include information on how to tune a controller while it is in operation and how to coordinate the application of different controllers based on the operating point of the system. For instance, in aircraft control, certain key variables are used in the tuning (scheduling) of control laws, and these may be exploited by a fuzzy controller in a unique approach to the design and implementation of a gain scheduler.

In process control, engineers or process operators often have considerable heuristic expertise on the tuning of PID controllers while they are in operation. This expertise may be codified and loaded into

the rule base of a fuzzy PID tuner to ensure that a PID controller is properly tuned at all times.

Another control application (this one within the author's personal experience) is the selection and tuning of controllers

for a two-link flexible robot [see diagram opposite]. With such a robot, the end of the arm will vibrate quite a bit if no feedback control is employed [see red response curve below]. Fuzzy and conventional controllers developed to reduce the



▲ The use of a rule-based supervisor [middle] to switch between two fuzzy controllers—one for coarse movements and one for fine—dramatically reduced the movement of the end of the robot arm [top, blue response curve] from the wild oscillations that occurred in the absence of feedback [top, red response curve]. With the fuzzy model reference learning control (FMRLC) [bottom], very similar results are obtained [top, black response curve] but with no need to manually construct a rule base. (Note the control system conventions: a diagonal arrow indicates that a controller is being tuned—not simply accepting input data. The inset box in the top drawing shows schematically the initial [dashed] and final [solid] positions of the robot arms.)

vibration worked reasonably well over the entire workspace of the robot; however, our experience indicated that control could be made more effective if a supervisor were added. That supervisor [middle drawing on previous page] would coordinate the action of two fuzzy controllers—one for large-angle movements and one for fine endpoint positioning—and would provide the tuning appropriate to each situation.

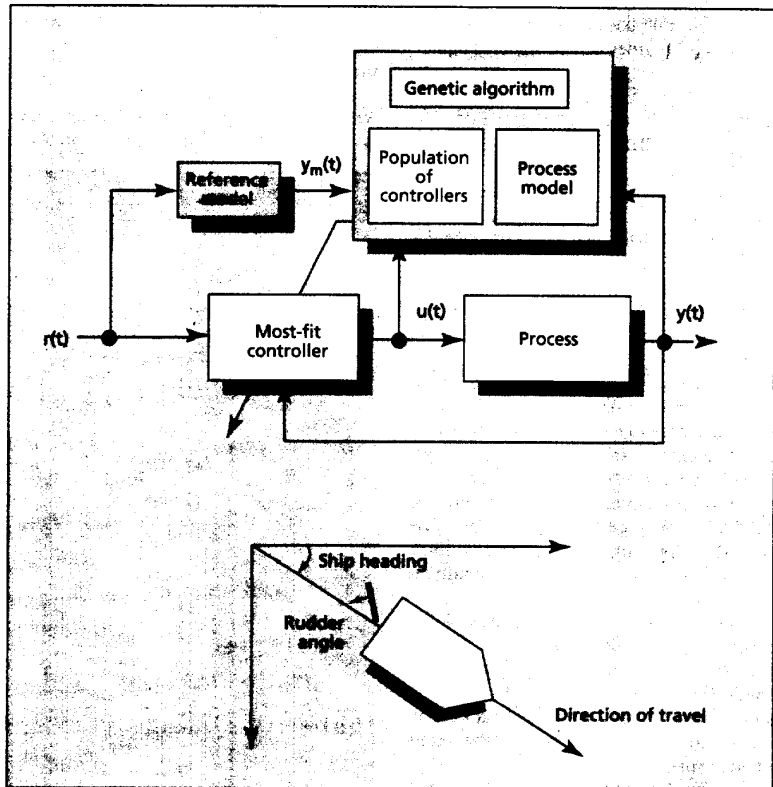
Accordingly, we added a supervisor with rules that indicated that one controller should be used when a link's endpoint was far from its destination and the other when it was near. The first controller was designed to move the robot without injecting energy into the flexible modes, while the second was designed for fine positioning so as not to amplify disturbances when the links were near their endpoints.

The rule-based supervisor simply switches between the two controllers depending on where the links are relative to their desired positions. The results it delivered were excellent [see blue response curve on previous page]—as well they should have been. After all, we had amassed years of experience in the design of conventional controllers for this testbed—some of which we could load into the rule base. Through this, and other experiments, we have found that intelligent control can sometimes capture valuable heuristic information, which may be difficult to incorporate into conventional controllers.

### Fuzzy learning control

In other fuzzy control approaches, the goal is to implement inductive, rather than deductive, systems. These systems not only can learn, but can also generalize from particular examples—for example, from the system's current behavior. Such approaches typically fall under the rubric of "fuzzy learning control" or "fuzzy adaptive control". One promising approach is called fuzzy model reference learning control (FMRLC). With it, a fuzzy controller is used in much the same way as in the previous discussion except that it begins with an empty rule base [bottom drawing on previous page]. A "reference model" with output  $y_m(t)$  is used to characterize the desired closed-loop system behavior—that is, it holds the performance specifications (for example, if the reference model is a simple first-order low-pass filter, then its time constant specifies how fast the closed-loop system should react to a step input). Then, a learning mechanism compares  $y(t)$ , the actual performance, with  $y_m(t)$  and decides how to synthesize or tune the fuzzy controller so that the difference between  $y(t)$  and  $y_m(t)$  goes to zero.

In particular, the learning mechanism



▲ Key to the operation of the genetic model reference adaptive controller (GMRLC) [top] is a genetic algorithm that maintains a population of controllers, which evolve in accordance to rules like those governing biological evolution. In a cargo ship steering application [bottom], it worked surprisingly well [see results on next page]. In this case,  $r(t)$  is the reference input (ideally desired heading),  $y_m(t)$  is the reference model output (practical desired heading),  $y(t)$  is the actual heading, and  $u(t)$  is the commanded rudder angle.

moves the centers of the output membership functions [of the rules previously applied to create the error between  $y(t)$  and  $y_m(t)$ ] so that the nonlinearity induced by the fuzzy controller will force the actual output,  $y(t)$ , to track the desired one,  $y_m(t)$ . The FMRLC has been successfully applied to the flexible robot described above, where it automatically synthesized a fuzzy controller rule base that achieved results [black response curve on previous page] comparable to those obtained in the rule-based supervisory control approach for which the rule bases were manually constructed. In addition, the FMRLC has been applied to the cruise control problem described above, a cargo ship steering application, anti-skid brakes, and reconfigurable control for aircraft, where it learns to compensate for failures in aircraft systems.

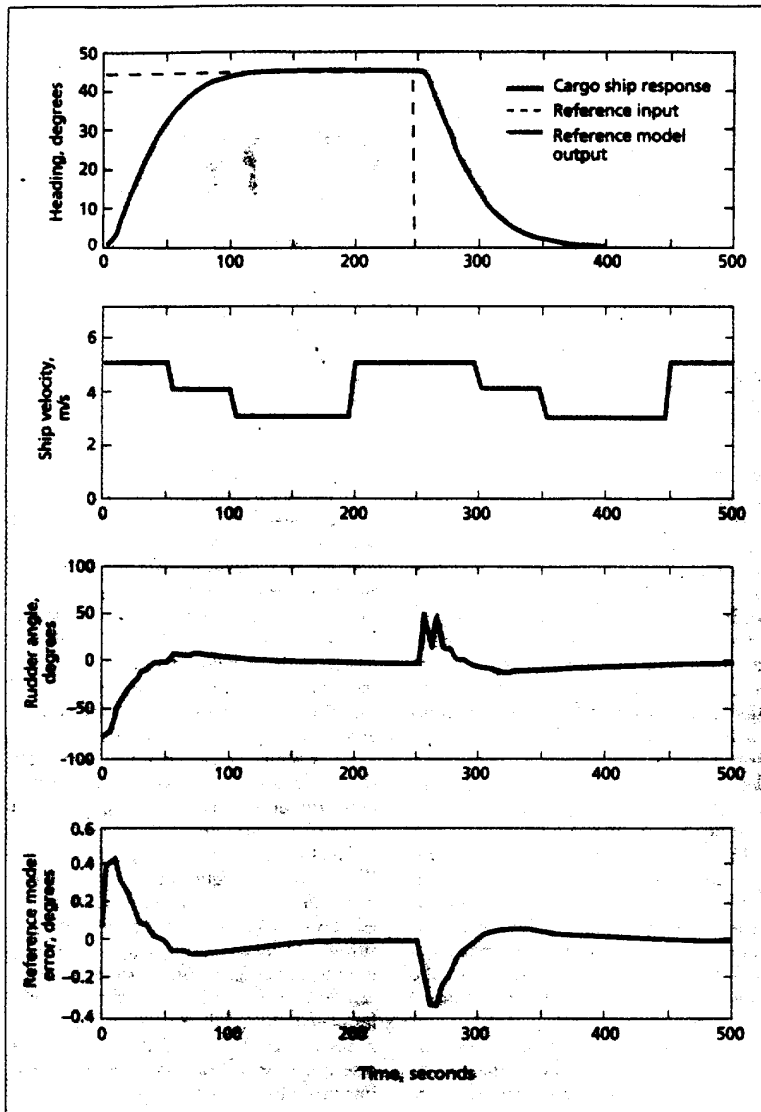
Our experiences with the FMRLC overall indicate that significant advantages may be obtained from controllers that can truly learn from their experiences (while forgetting useless information), so that when a situation is encountered repeatedly, the controllers will know how to

handle the later instances. This would appear to be an improvement upon those adaptive controllers that perform some adaptation every single time they encounter a situation, no matter how many times they have met it before.

### Knowledge-based control

While fuzzy control techniques similar to the kinds described above have been employed in a variety of industrial control applications, more general knowledge-based controllers have also been used successfully. For instance, expert systems (computer programs that roughly emulate the way an expert thinks through a problem) are being used in a supervisory role similar to the one illustrated on p. 57. Others, called expert controllers, have been put to work controlling complex processes. Such expert systems allow for more general knowledge representation techniques (not just fuzzy logic rules) and for inference mechanisms capable of implementing more complex reasoning strategies.

In addition, there are planning systems (computer programs that emulate the way



▲ The GMRAC works so well that the simulated cargo ship response overlies the reference model nearly perfectly [top] despite changes in the ship's speed during the maneuver [second trace from the top]. As would be expected, the rudder moves very little except to initiate the turn and to make adjustments immediately afterwards [third trace from the top]. The slight differences between the cargo ship response and the reference model may be examined in the expanded view [bottom trace].

experts plan) that have been used in path planning and for making high-level decisions about tasks for robots. Although these expert and planning system approaches certainly have promise, and have had some successes in selected applications, opportunities still exist for them to prove themselves in a wider context.

### Neural networks for control

Next to fuzzy-logic systems, probably no other intelligent control area has stirred as much interest as the application of artificial neural networks for control. In

applying these networks, engineers try to emulate the low-level biological functions of the brain to solve difficult control problems. For instance, for the inter-vehicle spacing control problem described earlier [p. 55], an artificial neural network may be trained to remember how to regulate the inter-vehicle spacing by being repeatedly supplied with examples of how to perform the task correctly.

After training, the neural network can be implemented on the vehicle to regulate the inter-vehicle distance by recalling the proper throttle input for each value of the

inter-vehicle distance (and rate of change of that distance) that is sensed. Other neural control approaches bear some similarity to the fuzzy model reference learning controller discussed above in that they automatically learn how to control a system by observing the system's behavior.

### Genetic algorithms for control

Yet a third approach to intelligent control is the one based on genetic algorithms. Here, the goal is to embody the principles of evolution, natural selection, and genetics from natural biological systems in a computer algorithm. Essentially, the genetic algorithm performs a parallel, stochastic, but directed search to evolve the most fit population.

It has been shown that a genetic algorithm can be used effectively in the (off-line) computer-aided design of control systems since they can artificially "evolve" an appropriate controller that meets the performance specifications to the greatest extent possible. To do this, the genetic algorithm maintains a population of strings that each represent a different controller (digits on the strings characterize parameters of the controller). It works on those strings with the genetic operators of reproduction, crossover, and mutation (representing respectively, the survival of the fittest, mating, and the random introduction of new "genetic material"), coupled with a fitness measure (which often involves the performance objectives) to spawn successive generations of the population.

After many generations, the genetic algorithm frequently produces an adequate solution to a control design problem. Its stochastic, but directed, search helps avoid locally optimal designs and seeks to obtain the best design possible.

Another, more challenging problem is trying to make controllers that evolve while the system is actually operating. Recently progress in this direction has been made by the introduction of the genetic model reference adaptive controller (GMRAC) [top part of drawing on preceding page]. As in the EMRLC, the GMRAC uses a reference model to characterize the desired performance. The key to the GMRAC is a genetic algorithm that maintains a population of strings representing the candidate controllers. That algorithm employs a model of the process, along with process data, to evaluate the fitness of each of the controllers at each time step.

Based on these fitness evaluations, the genetic algorithm propagates controllers into the next generation through the workings of the three standard genetic operators. In the end the controller that is deemed most fit is used to control the system. This allows the GMRAC to continually evolve a controller from one time

step to the next, and hence to constantly tune it in response to changes in the process (variations in temperature, aging of the plant, and so on) or in response to user changes of the reference model.

As an example, consider a cargo ship steering application [bottom part of drawing on p. 58]. In our simulation studies on that case, we seeded the initial population of controllers in the GMRAC with a random selection of gains for a proportional-derivative (PD) controller. Then we chose a reference model in accordance with the design specifications, embedded a simple linear process model in the GMRAC, and selected an appropriate fitness function. Finally, crossover and mutation probabilities of 0.6 and 0.24, respectively, were assigned, and the GMRAC was ready for a simulation run.

The results were quite impressive [see curves on preceding page]. Note that the cargo ship direction tracks the reference model output nearly perfectly despite the use of a nonlinear model of the plant and variations in the speed of the ship during the maneuver. (Lowering the speed makes it harder for the rudder to steer the ship.)

On the basis of our results and a reading of the literature [see *To Probe Further*, p. 62], we believe the GMRAC to be quite promising as a new technique for stochastic adaptive control. It provides a mechanism through which alternative controllers can be quickly applied to the problem when it looks as if they will be useful. Moreover, it has some inherent capabilities for learning through the evolution of its population of controllers.

Research activity in intelligent control typically focuses on algorithms based on fuzzy, expert, neural, or genetic principles hierarchically interconnected as demanded by different applications. Interconnecting diverse systems naturally raises the question of stability. While some recent work has focused on stability analysis of intelligent control systems, a significant amount of work remains to be done to establish a sound theoretical approach for the verification of the behavior of intelligent control systems.

In addition, there is a need to focus much more effort on detailed engineering cost-benefit analysis to study the advantages and disadvantages of intelligent control techniques and to determine whether they have anything to offer over conventional control approaches. Perhaps more importantly, there is a need to study very carefully whether and how intelligent control may be used to implement autonomous control systems.

#### Ultimate goal: autonomous control

Autonomous behavior has two basic ingredients: high performance and unassisted action. PID controllers can achieve

only certain performance levels and provide for only a limited amount of automation. More advanced controllers tend to offer higher levels of performance and greater automation capability.

A key characteristic of highly autonomous control systems is that they perform well under all process operating conditions and performance demands—even in the presence of failures. While autonomy may be achieved in a variety of ways, the focus here will be on doing so with intelligent control.

#### Intelligent autonomous controllers

One good way to look at a generalized intelligent autonomous controller is as a three-layer system having an execution layer, a coordination layer, and a management layer. The execution layer connects to the process under control via sensors and actuators. The management layer interfaces with other systems and with human operators. Information is processed and transmitted between those two layers through the coordination layer.

Each layer has a considerable degree of autonomy. The execution layer, for example, is not just a collection of data-acquisition and actuator hardware. It executes low-level numeric signal processing and control algorithms such as PID, optimal, adaptive, and intelligent controls; parameter estimators; and failure detection and identification (FDI) routines.

The coordination level tunes, schedules, supervises, and redesigns the execution level algorithms. It also handles crisis management, planning and learning capabilities for the coordination of execution level tasks, and higher-level symbolic decision-making for FDI and control algorithm management.

The management level supervises the lower level functions and manages the interfaces with humans and other systems. In particular, it interacts with the users in generating goals for the controller and in assessing the capabilities of the system. The management level also monitors the performance of the lower-level systems, plans activities at the highest level, and (in cooperation with human personnel), learns at a high level about the user and the lower-level algorithms.

Intelligent systems and controllers may be employed as appropriate in the implementation of various functions at any of the three levels of the intelligent autonomous controller. For example, adaptive fuzzy control may be used at the execution level for adaptation. Genetic algorithms may be used at the coordination level to pick an optimal coordination strategy. And planning systems may be used at the management level for sequencing operations.

Hierarchical controllers that mix intelligent and conventional techniques are commonly used in the intelligent control of complex dynamical systems. Because such systems demand a high degree of autonomy, they require a variety of decision-making approaches to handle the complex dynamical learning and reasoning involved.

Several fundamental characteristics have been identified for intelligent autonomous control systems [see *To Probe Further*, the first book listed]. In general, duties are delegated from the higher to the lower levels and the number of distinct tasks typically increases as we go down the hierarchy. Higher levels are often concerned with slower aspects of the system's behavior and with its larger portions, or broader aspects.

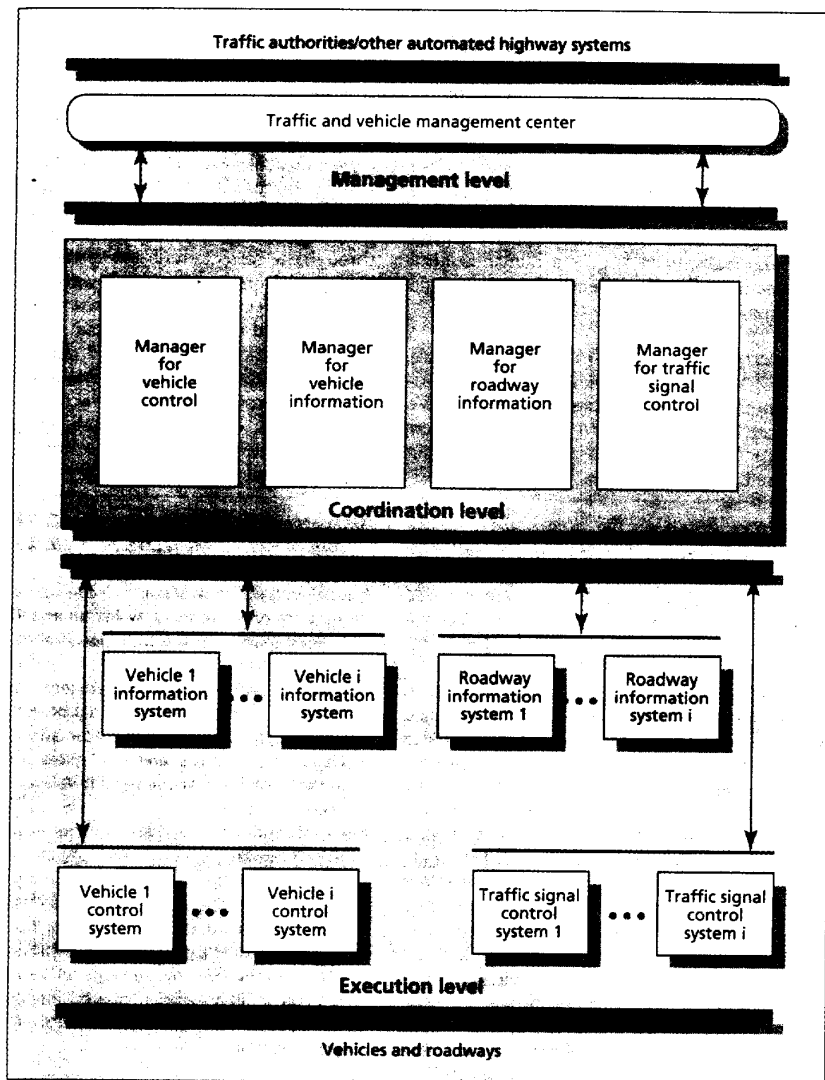
There is then a smaller contextual horizon at lower levels—that is, the control decisions are made by taking less information into account. Higher levels are typically concerned with longer time horizons and more extensive information. It is said that there is "increasing intelligence with decreasing precision as one moves from the lower to the higher levels" [see the second book in *To Probe Further* and its references]. At the higher levels there is typically a decrease in time-scale density, a decrease in bandwidth, or system rate, and a decrease in the decision (control action) rate.

Finally, note the ongoing evolution of the intelligent functions of an autonomous controller: by the time one is implemented, it no longer looks intelligent, just algorithmic. (It is this evolutionary principle, doubts about our ability to implement "artificial intelligence," and the fact that implemented intelligent controllers are nonlinear controllers that makes many researchers feel more comfortable focusing on enhancing autonomy rather than achieving intelligent behavior.)

#### Automated highway systems

To make the operation of autonomous systems and the notion of autonomy more concrete, let us examine the problem of automating a highway system. One possible general functional architecture for automated highway systems is shown in the diagram on the next page, which is clearly based on our three-level model. This example supposes that many vehicles are operating on a large roadway system in the metropolitan area of a large city. Each vehicle is equipped with a system that can control its brakes, throttle, and steering to automate the driving task (whether for normal operation or collision avoidance).

In addition, suppose that there is a vehicle information system in each vehicle that provides updated information to



The three-level organization of a generic intelligent autonomous controller is well illustrated in this system for an automated highway system. In general, the upper layers process a wider variety of information than the lower layers, and make decisions at a lower rate. For example, the management level might reconfigure traffic signaling because of changes in the weather. In steering an individual vehicle, the execution level, by contrast, would make many rapid decisions based on fairly little data.

both the driver and the overall system. (The driver would be given data on such subjects as the state of his vehicle's health, traffic congestion, road construction, accidents, weather, road conditions, and the location of hotels and restaurants. The overall system will be informed if the vehicle has had an accident or, for example, if its brakes have failed.)

The roadway is equipped with traffic signal controllers—for intersections and ramp metering, for example—and the roadway information systems that provide information to the driver and other subsystems. Among these other subsystems are automatic signaling systems that provide

re-routing information in case of congestion, poor road conditions, and so forth.

It is these four components that form the execution level in the intelligent autonomous controller. Of course, they will be physically distributed across many vehicles, roadways, and portions of the metropolitan area.

At the coordination level, there is a manager for vehicle control that may coordinate the control of closely spaced vehicles to form "platoons," which it then may maneuver to prevent collisions. The coordination layer may also provide information about such control activities to the rest of the system. In addition, the

coordination level has a manager for vehicle information, which makes sure that appropriate vehicles get the correct information about road, travel, and traffic conditions. That manager also manages and distributes the information on accidents and vehicle failures that comes in from vehicles so, for example, the control manager can navigate platoons to avoid collisions.

A third coordination-level manager, the one that handles traffic signal control, has two main functions. Using input from the roadway information system, it adapts the traffic light sequences at several connected intersections in such a way as to reduce congestion, and it reports on what it has done to the other subsystems (for example, the vehicle information systems).

Finally, the manager for roadway information tells the other subsystems about road conditions, accident information, and congestion. It also passes information from the other subsystems to the roadway for use by the changeable message signs—for example, re-routing information from the traffic signal control manager. As shown in the drawing [left], there are several copies of each of the managers and the entire coordination level as needed for different neighborhoods in the metropolitan area.

The management level deals with the broadest aspects of traffic flow. For one thing, it interfaces with other automated highway systems and with traffic authorities—in the latter case to inform police and emergency services about accidents and to gather information on construction, weather predictions, and major events that might affect traffic. For another, it advises traffic authorities on the best way to avoid congestion given current weather conditions, construction, and expected traffic loads. It also monitors the performance of all the subsystems in the coordination and execution layers and suggests corrective actions when necessary.

Notice that, in terms of the fundamental characteristics of intelligent autonomous control systems, duties are successively delegated as the controller hierarchy is descended. High-level tasks at the management level involve such activities as reconfiguring traffic signaling due to construction and weather. At the coordination level, the manager for roadway information and traffic signal control may develop new signaling strategies. And finally, that strategy would be implemented at the exe-

cution level on changeable message signs and the traffic signal control strategy.

In short, the higher levels of the hierarchy are typically concerned with slower and broader aspects of the system behavior (although in an accident the traffic and vehicle management center would react as quickly as possible to alert emergency vehicles). The lower levels of the system, conversely, take much less into account in reaching decisions, but tend to make decisions at a higher rate. Control corrections made as a vehicle steers around a curve may occur every few milliseconds, while the management center decisions may occur every few minutes or hours.

Clearly there is the need for a significant amount of interdisciplinary activity to implement such a complex control system. No single control technique (con-

ventional or intelligent) can be used to solve the diversity of problems found in a complex automated highway system. While conventional systems and control technologies will certainly find wide use in automated highway systems, it seems likely that intelligent systems and control techniques will prove to be useful for at least some functions, especially considering the focus on automating what has traditionally been largely a human control activity. Similar statements seem to hold for many other autonomous systems.

#### Where is this technology going?

The fields of intelligent and autonomous control are in their infancy. We are only beginning to find some answers to the questions posed in the opening remarks of this paper. While some "auto-

nomous" robots and vehicles have been implemented, there is still much room for improvement.

Current intelligent systems can only roughly model their biological counterparts, and hence, from one perspective, they can achieve relatively little. What will we be able to do if we succeed in emulating their functions much more completely? Achieve full autonomy through the correct orchestration of intelligent controls implemented with new computing technologies like neural networks? Could we achieve the same goals with conventional methods and conventional computing technology? Regardless of how we proceed, the goal of achieving autonomy is exciting and challenging, and is likely to produce many technological benefits along the way. ♦

#### To probe further

There are many general introductions to intelligent autonomous control. From Kluwer Academic Publishers, Norwell, Mass., come *An Introduction to Intelligent and Autonomous Control* (1993), edited by P. J. Antsaklis and the author of this article, and *Intelligent Robotic Systems: Theory, Design, and Applications* (1992) by K. P. Valavanis and G.N. Saridis. M. M. Gupta and N. K. Sinha are the editors of *Intelligent Control: Theory and Practice* (IEEE Press, Piscataway, N.J., 1995), as are D. A. White and D. A. Sofge of *Handbook of Intelligent Control: Neural, Fuzzy, and Adaptive Approaches* (Van Nostrand Reinhold, N.Y., 1992).

For other general theories on intelligent control, see J. S. Albus's "Outline for a Theory of Intelligence," *IEEE Transaction on Systems, Man, and Cybernetics*, Vol. 21, no. 3, 1991, and in the same publication, R. F. Stengel's "Toward Intelligent Flight Control," Vol. 23, no. 6, 1993.

An explanation of the relationships between traditional and intelligent control is given in "Bridging the Gap Between Conventional and Intelligent Control," by the author of this article. It appears in the "Special Issue on Intelligent Control," *IEEE Control Systems Magazine*, Vol. 13, no. 3, pp. 12-18, June 1993.

Also worth study is "Defining Intelligent Control—Report of the Task Force on Intelligent Control," by P. J. Antsaklis, its chair. The report appeared in *IEEE Control Systems Magazine*, Vol. 14, No.3, pp. 4-5, and pp. 58-62, June 1994.

An introduction to fuzzy systems and control is presented by L.X. Wang in *Adaptive Fuzzy Systems and Control: Design and Stability Analysis* (Prentice-Hall, Englewood Cliffs, N.J., 1994) and by this author with S. Yurkovich in "Fuzzy Control," in the *Handbook on Control*, edited by W. Levine (CRC Press, 1996).

More details on the fuzzy supervisory approach are provided by the last two authors plus V. G. Moudgal in "Rule-Based Control for a Flexible-Link Robot," *IEEE Transactions on Control Systems Technology* printed in Vol. 2, no. 4, pp. 392-405, December 1994.

An introduction to the FMRLC technique is contained in a paper by J. R. Layne and this author, "Fuzzy Model Reference Learning Control for Cargo Ship Steering," *IEEE Control Systems Magazine*, Vol. 13, no. 6, pp. 23-34, December 1993.

For details on the application of the FMRLC to the flexible robot, see V.G. Moudgal, W. A. Kwong, K. M. Passino, and S. Yurkovich in "Fuzzy Learning Control for a Flexible-Link Robot," *IEEE Transactions on Fuzzy Systems*, Vol. 3, no. 2, May 1995.

Two introductions to expert control and planning systems for control, besides the book by Antsaklis and Passino, are K. J. Astrom, J. J. Anton, and K. E. Arzen's "Expert Control," (*Automatica*, Vol. 22, pp. 277-86) 1986, and T. Dean and M. P. Wellman's *Planning and Control* (Morgan Kaufman, Calif., 1991).

Amid the many books on neural networks, there is the one edited by M.M. Gupta and D.H. Rao, titled *Neuro-Control Systems: Theory and Applications* (IEEE Press, Piscataway, N.J., 1994).

An excellent introduction to genetic algorithms is D.E. Goldberg's *Genetic Algorithms in Search, Optimization, and Machine Learning* (Addison-Wesley, N.Y., 1989).

More about genetic adaptive control is presented by L. Porter and this article's author in "Genetic Model Reference Adaptive Control," *IEEE International Symposium on Intelligent Control*, pp. 219-24, Columbus, Ohio, August 1994.

For greater detail about failure detection and identification systems, go to the book above by Antsaklis and Passino, or read H. Rauch's "Issues in Intelligent Fault Dia-

gnosis and Control Reconfiguration," *IEEE Control Systems Magazine*, pp. 6-12, Vol. 14, no. 3, June 1994.

More details on general learning control systems are presented in J. A. Farrell and W. Baker's chapter in the book listed above by Antsaklis and Passino.

For work on architectures for intelligent controllers, see other chapters in that book by J. S. Albus, B. P. Zeigler, L. Acar and U. Ozguner, A. H. Levis, and A. Meystel, and in the book listed above by K. P. Valavanis and G. N. Saridis.

More information on architectures for automated highway systems (AHS) appears in "ITS Architecture Development Program Phase I Summary Report," *ITS America*, U.S. Department of Transportation, November 1994. For a general overview of this area, see the *IEEE Transactions on Vehicular Technology*, Special Issue on Intelligent Vehicle Highway Systems, Vol. 40, February 1991.

#### Acknowledgment

The author gratefully acknowledges the partial support of the National Science Foundation (Grants IRI-9210332 and EEC-9315257) and the Center for IVHS at The Ohio State University (IVHS-OSU).

#### About the author

Kevin M. Passino is with the Department of Electrical Engineering at The Ohio State University. He is an associate editor for the *IEEE Transactions on Automatic Control*, is on the editorial board of the *International Journal of Engineering Applications of Artificial Intelligence*, is a guest editor for a special track on *Intelligent Control for IEEE Expert Magazine*, and serves on the Board of Governors of the IEEE Control Systems Society. He is also the general chair of the 1996 IEEE International Symposium on Intelligent Control.