When and Where Will AI Meet Robotics?

Issues in Representation

Ruzena Bajscy and Edward W. Large

Because perception-action systems are necessarily constrained by the physics of time and space, robotocists often assume they are best described using differential equations, a language that is specialized for describing the evolution of variables that represent physical quantities. However, when it comes to decision making, where the representations involved refer to goals, strategies, and preferences, AI offers a diverse range of formalisms to the modeler. However, the relationship between these two levels of representation—signal and symbol—are not well understood. If we are to achieve success in modeling intelligent physical agents, robotics and AI must reach a new consensus on how to integrate perception-action systems with systems designed for abstract reasoning.

In the early days of AI, robotics was an integral part of our research effort. All our major AI laboratories had research programs in robotics in the late 1960s and early 1970s. However by the 1980s, robotics had taken its own course separate from the core activities of AI. One might argue that such a split was inevitable, a natural result of specialization in a rapidly growing and maturing field such as ours, but in our pursuit of rational models of the mind, do we dare leave the body behind?

What is responsible for the divergence between these two fields that once were so intimately intertwined? Can AI and robotics ever be reunited, and if so, what would a new partnership look like? At the core, we believe, is the ubiquitous issue of representation. There is an enormous difference between dealing with physical systems that operate in our everyday environment and software systems that reside in various abstract worlds. This gap has led to divergence in many areas, including the following:

The problems: Robotics problems entail systems and agents interacting with the physical world, but AI deals mostly with abstract problems that lend themselves to symbolic representations.

The environment: Roboticists seek to design systems that function in physical environments that are always changing and intrinsically unpredictable. Software agents generally operate in human-designed worlds where one can have some measure of control over change or at least an a priori knowledge of the possibilities.

The tools: AI more commonly uses discrete mathematics, but robotics and machine perception make use of continuous mathematics. These tools also differentiate the typical educational background that characterizes the two areas: AI has more computer scientists, but robotics has more electrical and mechanical engineers.

The evaluation criteria: AI researchers seem to value novelty more, solving "hard" problems, showing existence of solutions, and so forth. Robotics, however, follows traditional engineering evaluation criteria: efficiency, reliability, accuracy of performance, and economy of the solution.

We admit that these divisions might be overexaggerated because in both fields, one can find counterexamples to the previous statements. Nevertheless, each of these points speaks to the differences one encounters when dealing with corporeal agents in the physical world versus software agents in cyberspace. Robotics concentrates most of its resources on modeling perception and action. Often, differential equations are used to embody relatively simple strategies for controlling hardware effectors based on sensory information. AI, however, emphasizes planning and abstract

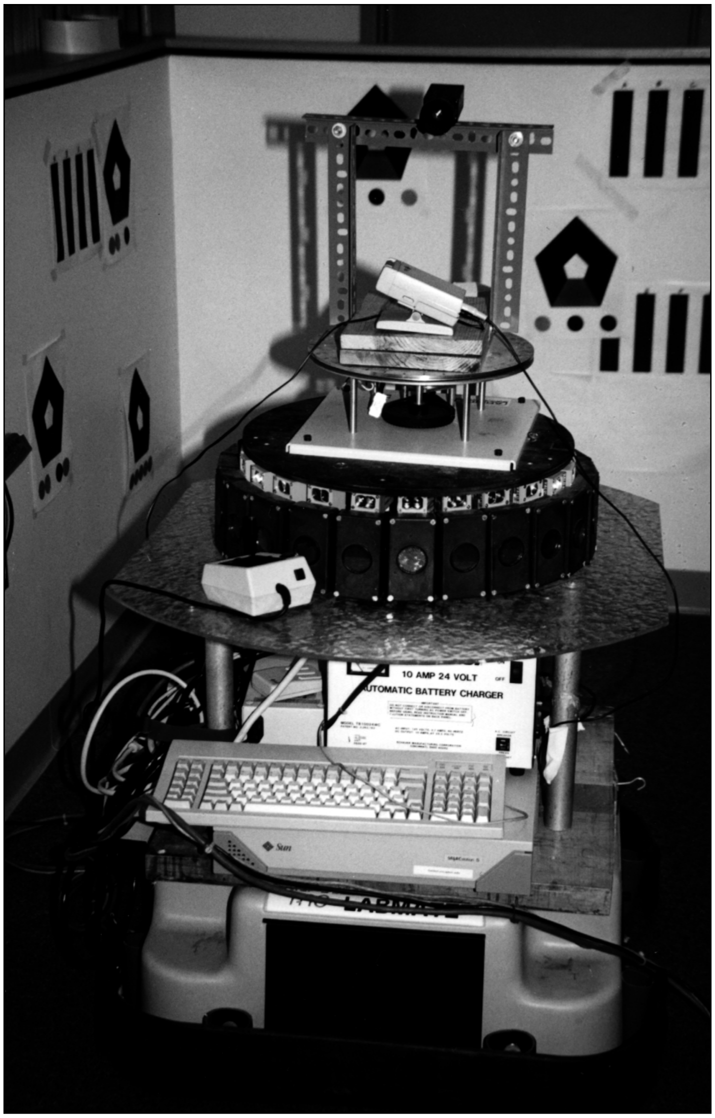

Figure 1. The Autonomous Agent.

The TRC platform serves as a mobile base. A stereo camera pair mounted on the front of the rig is used for obstacle detection. A third camera mounted on a pan platform is used for target detection and tracking.

reasoning. For example, logical, grammatical, or other discrete formalisms are used to model the complex operations involved in winning a chess match or parsing a sentence.

What seems clear is that as robotic agents are called on to perform increasingly complex tasks, they will be required not only to react flexibly in dynamically changing environments but also to make decisions, reason abstractly, and change perceptual or behavioral strategies. Conversely, as intelligent software agents are required to operate more and more on human terms, responding to sensory information and interacting physically with humans, they will be required to integrate more sophisticated perception-action capabilities with their abstract reasoning abilities. In addition, although each of these areas has been well studied, in robotics and AI respectively, the integration of perception-action systems with reasoning systems is less well understood. Thus, there is a great need to reconsider the relationship between AI and robotics.

The Problem of Representation

We will explore the issue that we believe is key to this relationship, the issue of representation. Representation is critical, especially when one considers how to find a description that is compact, yet expressive enough to enable the modeling of intelligent physical agents. What kind of mathematical tools are available to us? At the signal level, modeling of perception-action (reactive) behaviors can be anchored in differential equations and control theory, both linear and nonlinear. At the symbol level, higher-level control derives its models either from geometry (typically used in robotics) or from logics and rule-based systems such as are favored in the AI planning community. If time needs to be explicitly accounted for, then there are other tools available. At the signal level, time is implicit in the model of the reactive behaviors (for example, using differential equations). At the symbol level, discrete states are generally considered, and these can be modeled using discrete-event systems; temporal logics; and, at an even higher level, fluents. If uncertainty and disturbances must be modeled, then one must bring to bear stochastic models and probability theory; partially observable Markov decision models is one such example. Finally, utility functions and cost-benefit trade-offs come to play in conjunction with game theory, optimization, selection of strategies, and complexity considerations. Examples of such approaches as they have been applied to robotics are presented in Alami et. al. (1998); Clementia, Di Felice, and Hernandez (1997); Cohn (1995); Russell and Subramanian (1995); and Sandewall (1994). Hybrid system approaches such as discussed in Arkin (1998), Brockett (1993), Dickmanns (1997), Nagel and Haag (1998), Nerode and Remmel (1991), and Ramadge and Wonham (1987) combine discrete systems with lower-level control systems. However, the use of all these mathematical tools is predicated on the assumption that label assignment (the segmentation of the sensory or control signals) is performed externally to the system.

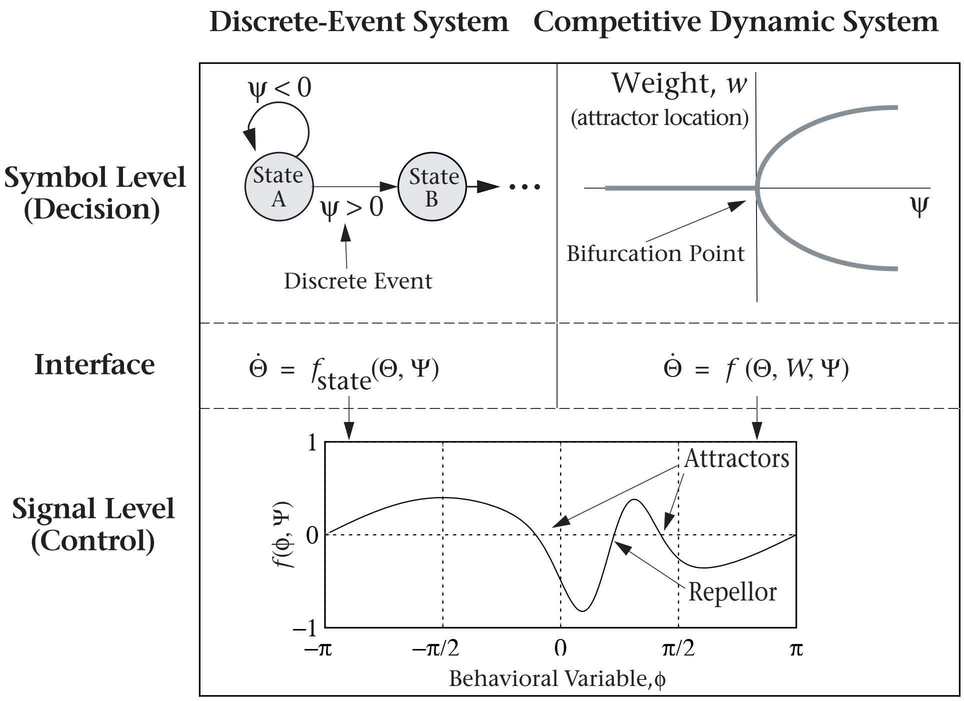

Figure 2. Two Approaches to Decision Making and Behavioral Sequencing.

In the discrete-event system, the discrete states of a finite-state machine correspond to control of perception-action by a distinct control law ($\dot{\Theta} = F_{state}(\Theta, \Psi)$). Transitions between states are governed by guard conditions on perceptual variables (for example, ψ). In the competitive dynamic system, each state variable (for example, w) controls the weighting of a task constraint at the signal level. Behavior is shaped directly by the competitive dynamic system, $\dot{\Theta} = F(\Theta, W, \Psi)$, as the symbol-level system activates and deactivates attractor and repellor contributions to the behavioral dynamics. As perceptual parameters (for example, ψ) change, bifurcations cause qualitative changes in perception-action behavior.

Signal and Symbol

In our view, the main unsolved problem is how to segment the continuous signal into states, strategies, labels, and so forth, and how to arbitrate among states for effective interaction with the environment (for example, Shi and Malik [1998], Tari, Shah, and Pien [1997], and Large, Christensen, and Bajscy [1999]). In the GRASP Laboratory, we have concerned ourselves with the problem of representation in signal-symbol systems over the past several years. One approach to this problem involves the study of *intelligent physical agents,* agents that can operate in the physical world with all its uncertainty yet behave intelligently, making decisions about how best to perform simple and complex tasks in a range of real–world environments. A picture of one such agent is shown in figure 1. It consists of a TRC LABMATE mobile platform equipped with a stereo camera pair used for obstacle detection; a third camera mounted on a turntable for visual tracking; and several computers used for processing sensory signals, generating control signals, and making decisions.

The control system that models perception-action behavior transforms visual input into control signals to enable the physical agent to carry out various navigation tasks. We use a dynamic system approach proposed by Schoner, Dose, and Engels (1996). In this approach, behavior is controlled by differential equations that describe the rate of change of *behavioral variables*, such as heading direction and velocity. At any instant in time, the values of these variables describe the agent's behavior. Over time, the dynamic system generates a series of values, controlling the behavior of the agent. Our dynamic system has the form

$$D\Theta/dt = F(\Theta,\Psi) \qquad (1)$$

where $\Theta = [\phi\ \upsilon]^T$ is a vector of behavioral variables, heading direction, and velocity and is a vector of variables that represent perceptual information, such as direction to the target and size of obstacles. An example of such a function is shown in figure 2 (bottom panel). Three *fixed points* can be seen in the figure as points where the value of $f(\Theta,\Psi)$ is zero (that is, $d\phi/dt = 0$, so heading direction is fixed). If the slope of the function around a fixed point is positive, the value of the behavioral variable is pushed away from this value by the action of equation 1; such an unstable fixed point is called a *repellor*. If the slope of the function is negative, it is a stable fixed point, called an *attractor*, because the behavioral variable is pulled toward this value by the differential equation.

The behavior of the agent is controlled by the configuration of attractors and repellors: Desired actions (such as moving toward a target) are modeled as attractors, and undesired actions (such as moving toward an obstacle) are modeled as repellors of the perception-action dynamic system. *Task constraints* determine the mapping from perceptual information to behavioral attractors and repellors. If the task is to go to the desk, the action of moving toward the desk is modeled an attractor, but other objects are considered obstacles (modeled as repellors). However, if the task is to rendezvous with another agent, then the action of moving toward the other agent is an attractor, and the desk is treated as an obstacle, and moving toward it is to be avoided. Thus, viewed as a representation of a perception-action behavior, this dynamic system incorporates task knowledge and makes use of perceptual information.

As the values of the perceptual variables change, the attractor-repellor layout changes. For example, if a target moves, the location of the corresponding attractor will move as well, thus providing behavioral flexibility—behavior adapts to accommodate changes in the environment. An even greater measure of flexibility is provided by *bifurcations* in the dynamic system—qualitative changes in the layout of attractors and repellors caused by changes in parameter values. For example, when a new obstacle comes into view, a repellor forms where there was no repellor before.

Although the dynamic control system provides a great deal of flexibility, it can only model one relatively simple perception-action behavior at a time. Complex tasks, however, typically require the execution of sequences of behavior. For example, one agent might need to rendezvous with an agent and then proceed toward a target location. To perform such tasks in a complex environment requires sequencing several simpler perception-action strategies. Building on the previous signal-level representation, we have investigated two approaches to modeling decision making and sequence generation: (1) a *discrete-event–system* approach and (2) a *competitive dynamic system* approach. There are three key differences between these two approaches: (1) the model of how the symbol level interfaces with the signal level, (2) the model of how perception is integrated into the decision-making process, and (3) the model of how decisions are captured at the symbol level.

Although the dynamic control system provides a great deal of flexibility, it can only model one relatively simple perception-action behavior at a time. Complex tasks, however, typically require the execution of sequences of behavior.

Before describing each approach in detail, we summarize the differences between the two systems in figure 2. In the *discrete-event– system* approach (Kosecka 1996), the symbol level interfaces with the signal level by realizing distinct behaviors as separate dynamic systems (that is, f_{state}, middle left panel); these are conceived of as elementary perception-action strategies. At any particular time, the behavior of the agent is governed by one of these equations. Decision making and sequencing are modeled at the symbol level using a finite-state machine (FSM) (top left panel). Individual FSM states correspond to elementary signal-level behaviors; when in a particular state, behavior is governed by a corresponding signal-level dynamic system. The arcs linking states are labeled with discrete events (for example, $\alpha > 0$) that summarize perceptual information. The perceptual system generates these events, which correspond to qualitatively different conditions in the environment. The occurrence of a specific event causes the switch from one state to another, modeling the decision to execute a different perception-action behavior. Thus, sequences of behavior are generated by traversing the arcs, which is, in turn, governed by the conditions on the current situation.

The *competitive dynamic system* model is formulated entirely within the qualitative theory of dynamic systems. Both signal-level control and symbol-level decision making are modeled using differential equations (Large, Christensen, and Bajscy 1999). However, the competitive dynamic system interfaces with the signal level differently than the discrete-event system. Rather than defining multiple elementary perception-action behaviors, a single master equation is defined containing all possible task constraints (middle right panel, figure 2). Then, each variable in the symbol-level system controls the weighting of a task constraint at the signal level, such that the symbol-level sys-

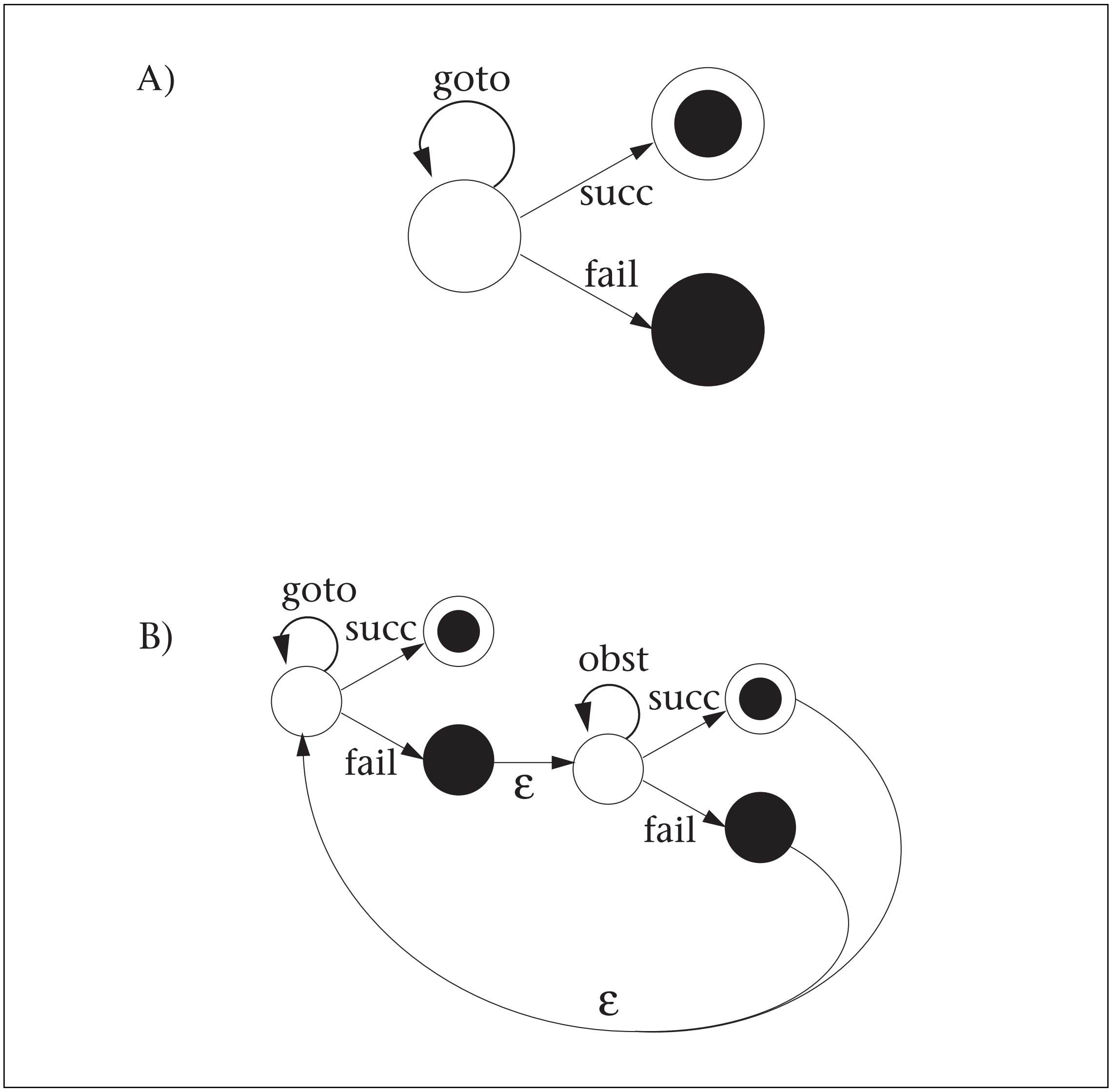

Figure 3. Finite-State Machine (FSM) *Models for Simple and Complex Behaviors.*

A. An FSM for elementary behavior GoTo. The control law (f_{GoTo}) is repeatedly invoked in the next state until successful (arrival at the goal) or unsuccessful (for example, detection of a spurious attractor) termination.

B. Finite-state model for a navigation behavior. Failure of GoTo is followed by the elementary behavior Escape. Once the agent clears the obstruction, GoTo is invoked again. This more complex behavior is able to handle a large variety of navigation situations.

tem can activate and deactivate attractor and repellor contributions to the behavioral dynamics. Different behaviors are modeled as fixed points (stable weight configurations) in the competitive dynamic system. The environment determines the values of the system parameters. As the perceptual information changes, parameters change, causing bifurcations in the symbol-level system (top right panel, figure 2), modeling the decision to cease executing one behavior and execute another instead. This approach differs from the discrete-event approach because the properties of dynamic systems, such as stability, bifurcations, and hysteresis, govern decision making and sequence generation.

Discrete-Event Systems

The discrete-event approach models elementary perception-action strategies as behavioral atoms, and each elementary control law is associated with a state in a simple finite-state machine. We define the composition operators for the FSMs by imposing some additional structure. The set of final states of an elementary behavior is partitioned into a set of successful and unsuccessful final states. By utilizing these primitives, it is possible to build

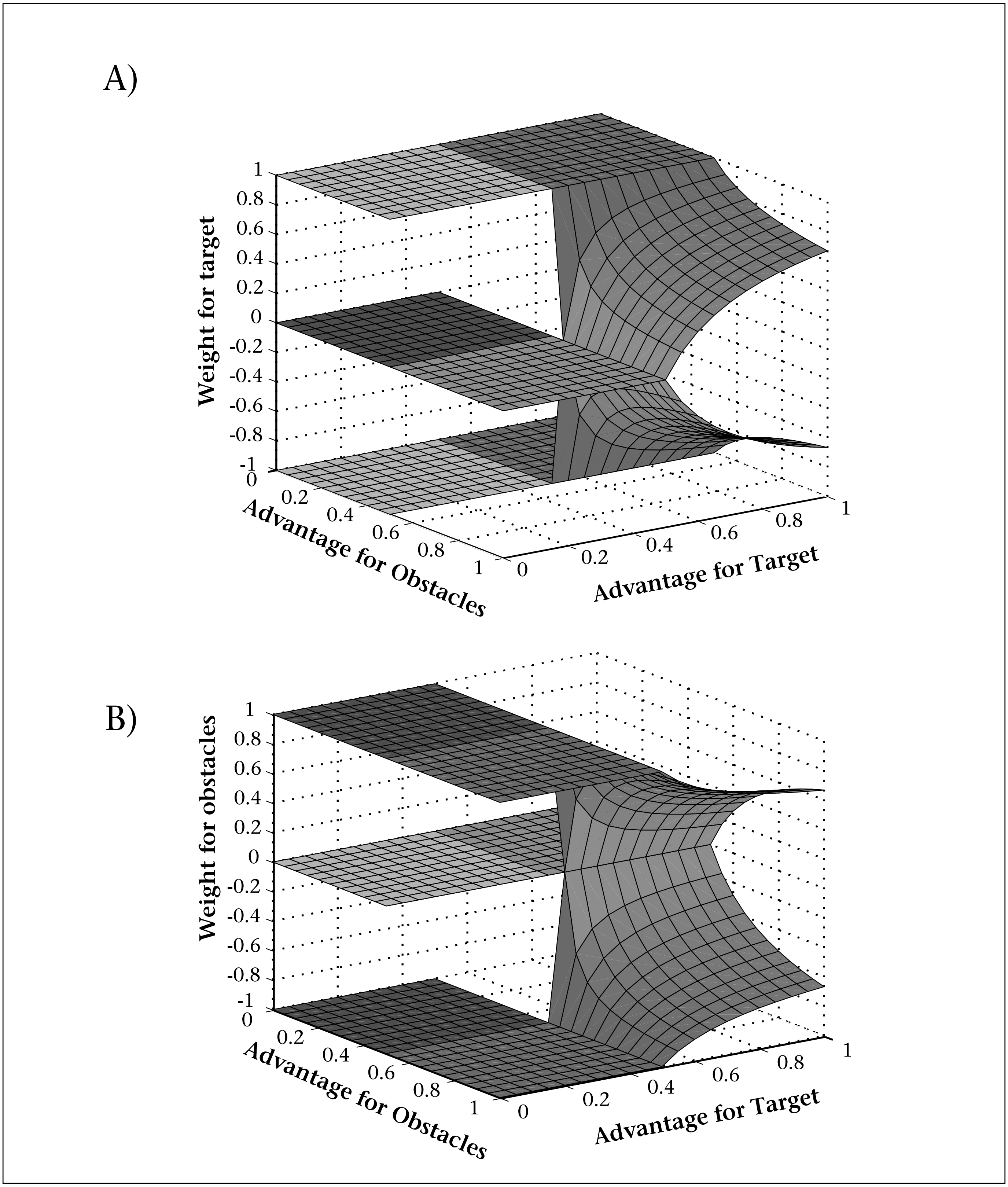

Figure 4. Bifurcations in a Competitive Dynamic System for Decision Making.

Two bifurcation diagrams are shown, one corresponding to each dimension of a two-dimensional system for system navigation. It is assumed that $\gamma_{tar,obs} = \gamma_{obs,tar} = 0.5$. A. The state variable w_{obs} determines the weighting of f_{tar}. B. The state variable w_{obs} determines the weighting of f_{obs}. Four qualitatively different behaviors, corresponding to four fixed points of the competitive dynamics, are shown.

models of more complex tasks, which are described as sequences of behavioral atoms. By using a task-specification language, complex FSMs are synthesized by sequencing simpler automata.

The FSM model of an elementary GoTo strategy is shown in figure 3a. A signal-level perception-action behavior is repeatedly invoked in the next state until the agent reaches the desired target location and makes a transition to the final state. If the strategy fails, the transition to the unsuccessful final state is made. As an example of a composition of elementary behaviors, consider the problem of moving to a target location while avoiding obstacles. In simple environments, one elementary perception-action behavior, GoTo, might do the trick. However, in complex environments with multiple obstacles arrayed in difficult configurations, our agent might get stuck in an area and never reach the target location (Large, Christensen, and Bajscy 1999). We address this problem by adding an Escape behavior that enables the agent to find its way out of enclosures and other spatial traps. The FSM for this behavior is shown in figure 3b. We assume that the fail signal to the GoTo behavior is generated whenever the agent detects an enclosure from which it must escape. When the fail signal is generated, the FSM enters the unsuccessful final state, and a transition is made to the initial state of Escape. When Escape terminates (when the agent has cleared the obstruction), the GoTo behavior resumes. The navigation task terminates successfully when the agent reaches the target location.

Competitive Dynamic Systems

The competitive dynamic system strategy models individual behaviors as the stable fixed points of a decision-making dynamic system. This system interacts with the signal level not by invoking separate elementary behaviors but by directly shaping perception-action strategies. The variables of the competitive dynamic system determine the weighting of the task constraints in the behavioral dynamic system. This interface between the two levels allows the decision-making system to activate and deactivate attractors and repellors in the perception-action system, synthesizing control laws on the fly. Thus, qualitatively different configurations of the weights give rise to distinct perception-action behaviors. In addition, distinct weight configurations, which arise as attractors in the competitive dynamic system, are functionally equivalent to the FSM states of the discrete-event system.

Decisions are made through bifurcations in the competitive dynamic system, and as with any dynamic system, bifurcations are caused by changes in the values of the system parameters. The decision-making system uses two types of parameter: (1) competitive advantage and (2) competitive interaction. These parameters are tied to perceptual information, so that decisions are made on the basis of the environment as sensed by the agent. First, each weight has an associated *competitive advantage* that describes whether the corresponding task constraint (for example, move toward target, avoid obstacles) is appropriate to the agent's current situation. For example, if obstacles are nearby, the Obstacles constraint will have a strong competitive advantage, but if the target is also in view, the Target constraint also has a strong advantage. The activation of both constraints simultaneously corresponds to the elementary GoTo behavior of the discrete-event system earlier. *Competitive interaction* describes the extent to which each constraint is consistent or inconsistent other constraints. For example, if the agent finds itself enclosed in an area with the target just beyond, the competitive interaction between the Obstacles and the Target constraints would increase so that the Target constraint would be deactivated temporarily, allowing the agent to escape from the enclosure. Deactivation of the target constraint corresponds to the Escape behavior of the discrete-event system.

We can understand in detail how these parameters interact to determine the behavior of the agent by constructing a bifurcation diagram such as that in figure 4. The bifurcation diagram shows the layout of fixed points for a two-dimensional system (that is, weights for the Target and Obstacles constraints) as a function of the perceptual parameters. In the figure, the competitive advantage parameters are varied from 0 to 1, assuming that the two competitive interaction parameters remain fixed at 0.5. Four qualitatively different regions (and, thus, behaviors) are pictured. Activation of both constraints corresponds to the GoTo behavior described earlier, but activation of the Obstacles constraint only corresponds to the Escape behavior.

Beginning in the front left corner of the parameter space, only the Obstacles constraint contributes to the behavioral dynamics (the Escape behavior). Moving to the right, as Target's advantage increases beyond 0.5, both Target and Obstacles constraints contribute to shape perception-action behavior (the GoTo behavior). As we next decrease the advantage of Obstacles, moving to the back left region, Obstacles is deactivated, but Target is activat-

Intelligent agents must be capable of bringing to bear a rich variety of perception-action strategies but, at the same time, reasoning and solving problems to perform both familiar and unfamiliar tasks in novel environments.

ed. If we once again decrease Target, moving toward the back left region of the parameter space, we notice that Obstacles remains inactive, but Target remains active. This last region of the state space is different from the others. Two behaviors are stable in this region. However, the system can only occupy one of these states at any given time. In this case, the state of the system is determined by its recent history, a phenomenon known as *hysteresis*. Finally, the boundaries of the four regions are determined by the values of the competitive interaction parameters. When competitive interactions change, the relative sizes of the different stable regions change as well. Each of these different regions corresponds to the execution of a qualitatively different perception-action behavior.

Comparing Approaches

We tested the two systems described previously to evaluate their relative performance in autonomous navigation tasks (Large, Christensen, and Bajscy 1999). We vary environmental complexity by constructing environments with different numbers of obstacles arranged in various configurations. We also vary task complexity; for example, a single agent performs simple navigation, or a pair of agents performs a cooperative task. In a range of tests, dynamic agents perform tasks faster and more reliably than discrete-event agents. They are able to maintain higher mean velocities, finding targets faster and with lower failure rates than discrete-event agents. However, the discrete-event agents also have advantages. They obey task constraints more faithfully, reluctant to relax constraints regardless of environmental complexity. These differences are the result of the model of decision making. The symbol-level dynamic system changes state less often than the discrete-event system, especially in the face of a noisy sensor reading. Although our experiments are not yet conclusive, by comparing modeling strategies in a careful way, we are gaining important insights into the special requirements of systems that must manipulate both signals and symbols at the same time and toward the same goal.

Conclusions

Early in the history of AI, many researchers came to believe that perception and action could be modeled by relatively simple transduction mechanisms, and therefore, abstract reasoning and problem solving were the difficult issues worthy of study. More recently, it has been argued that complex representation and reasoning might be unnecessary because many apparently intelligent behaviors can be modeled as perception-action systems situated in the physical world. Unfortunately, we have come to view both points of view as somewhat simplistic. Intelligent agents must be capable of bringing to bear a rich variety of perception-action strategies but, at the same time, reasoning and solving problems to perform both familiar and unfamiliar tasks in novel environments.

In this regard, the study of intelligent physical agents and their behavior is of tremendous theoretical and practical significance in AI. Not only are there a vast number of real-world applications where autonomous agents can be useful, but models of intelligent physical agents can serve as valuable starting points for theories of intelligent biological systems. The question that we ask is how to integrate models of perception-action behavior with models of problem-solving behavior.

Although we do not yet have the answer, we have two requirements for any solution: First, the description of the physical agent should take place within a structured framework that supports both analysis and theory making. Thus, we are allowed to develop design methodologies for artificial agents as well as develop rational theories of biological agents. Furthermore, any methodology that we propose should be compositional, allowing manageable and flexible system design through decomposition of complex problems or behaviors into subparts. Both of our systems meet these requirements. Finally, it is necessary to carefully compare the assumptions brought to bear by different strategies as we learn to model intelligent behavior in the real world.

What are the special requirements of systems that must interact with the physical world and also reason and solve problems? It is this question that must be addressed before we can claim a theory of intelligent physical agents and before AI and robotics can be reunited.

What are the special requirements of systems that must interact with the physical world and also reason and solve problems? It is this question that must be addressed before we can claim a theory of intelligent physical agents and before AI and robotics can be reunited.

References

Alami, R.; Chatila, R.; Fleury, S.; Ghallab, M.; and Ingrand, F. 1998. An Architecture for Autonomy. *International Journal of Robotics Research* 17(4): 315–337.

Arkin, R. C. 1998. Behavior-Based Robotics. Cambridge, Mass.: MIT Press.

Brockett, R. W. 1993. Hybrid Models for Motion Control Systems. In *Essays in Control: Perspectives in the Theory and Its Applications,* eds. H. L. Trentelman and J. C. Willems, 29–53. Boston: Birkhauser.

Clementini, E.; Di Felice, P.; and Hernandez, D. 1997. Qualitative Representation of Positional Information. *Artificial Intelligence* 95(2): 317–356.

Cohn, A. G. 1995. The Challenge of Qualitative Spatial Reasoning. *Computing Surveys* 27(3): 323–327.

Dickmanns, E. D. 1997. Vehicles Capable of Dynamic Vision: A New Breed of Technical Beings? In Proceedings of the Fifteenth International Joint Conference on Artificial Intelligence. Menlo Park, Calif.: International Joint Conferences on Artificial Intelligence.

Koenig, S., and Simmons, R. G. 1998. XAVIER: A Robot Navigation Architecture Based on Partially Observable Markov Decision Process Models. In *AI and Mobile Robots,* eds. D. Kortenkemp, R. P. Bonasso, and R. Murphy. Cambridge, Mass.: MIT Press.

Kosecka, J. 1996. A Framework for Modeling and Verifying Visually Guided Agents: Design, Analysis, and Experiments, GRASP TR 402, Ph.D. dissertation, School of Engineering and Applied Science, University of Pennsylvania.

Large, E. W.; Christensen, H. I.; and Bajscy, R. 1999. Scaling the Dynamic Approach to Path Planning and Control: Competition among Behavioral Constraints. *International Journal of Robotics Research* 18(1): 37–58.

Nagel, H. H., and Haag, M. 1998. Bias-Corrected Optical Flow Estimation for Road Vehicle Tracking. In *Proceedings of the International Conference on Computer Vision (ICCV '98),* 1006–1011, New Delhi, India: Narosa.

Nerode, A., and Remmel, J. B. 1991. A Model for Hybrid Systems. Paper presented at the Hybrid Systems Workshop, 17–19 May, Ithaca, New York.

Ramadge, P. J., and Wonham, W. M. 1987. Supervisory Control of a Class of Discrete Event Processes. *SIAM Journal on Control and Optimization* 25(1): 206–230.

Russell, S. J., and Subramanian, D. 1995. Provably Bounded-Optimal Agents. *Journal of AI Research* 2(1): 575–609.

Sandewall, E. 1994. *Features and Fluents: The Representation of Knowledge about Dynamical Systems.* Oxford, U.K.: Oxford University Press.

Schoner, G.; Dose, M.; and Engels, C. 1996. Dynamics of Behaviour: Theory and Applications for Autonomous Robot Architectures. *Robotics and Autonomous Systems* 16(4): 213–246.

Shi, J., and Malik, J. 1998. Self-Inducing Relational Distance and Its Application to Image Segmentation. In *Proceedings of the 1998 European Conference on Computer Vision, Volume 1,* eds. H. Burkhardt and B. Neumann, 528–543. Lecture Notes in Computer Science. Berlin: Springer Verlag.

Steinhage, A., and Schoner, G. 1998. Dynamic Systems for the Behavioral Organization of Autonomous Robot Navigation. In *Sensor Fusion and Decentralized Control in Robotic Systems: Proceedings of the International Society for Optical Engineering,* eds. P. S. Schenker and G. T. McKee, 169–180. Bellingham, Wash.: International Society for Optical Engineering.

Tari, S.; Shah, J.; and Pien, H. 1997. Extraction of Shape Skeletons from Gray-Scale Images. *Computer Vision Image Understanding* (Special Issue on Biomedical Image Analysis) 66:133–146.

Ruzena Bajcsy obtained her first Ph.D. in EE from Slovak Technical University (Czechoslovakia) in 1967; she was the first woman in Slovakia ever to obtain a Ph.D. She then went to Stanford University and obtained her second Ph.D. in 1972, studying AI under John McCarthy. Her dissertation topic was machine perception, and she wrote one of the first programs enabling recognition of textured patterns.

She then joined the faculty at the University of Pennsylvania, where she has continued her work on machine perception and computer vision, characterizing and solving problems involving segmentation, three-dimensional vision, and other sensory modalities that function together with vision (for example, touch). Bajcsy's General Robotic Active Sensory Perception (GRASP) Laboratory at the University of Pennsylvania is internationally recognized in the field.

She chaired her department from 1984 through 1986 and has served on numerous National Research Council and National Science Foundation (NSF) advisory boards and committees. She was elected a fellow of the Institute of Electrical and Electronics Engineers in 1992 and the Association of Computing Machinery in 1995; she is also a founding member of the American Association for Artificial Intelligence. In 1995, she was elected as a member of the National Institute of Medicine, and she became a member of the National Academy of Engineering in 1997. She has served for three years on the CRA board.

Bajcsy is currently the assistant director of the NSF Directorate for Computer and Information Science and Engineering. Her e-mail address is bajcsy@central.cis.upenn.edu.

Edward Large is an assistant professor at the Center for Complex Systems at Florida Atlantic University. He received his Ph.D. from The Ohio State University in 1994. He has held fellowships in cognitive science and psychology at the University of Pennsylvania and in AI at Toshiba's Research and Development Laboratory in Kawasaka, Japan. His research focuses on the dynamics of human perception, action, and cognition and the design of autonomous agents using dynamic principles. His areas of specialization include music perception, auditory perception, and robotics. His e-mail address is large@walt.ccs.fau.edu.


Software
Agents