To appear in the 1999 AAAI Spring Symposium Series: Hybrid Systems and AI

A Hybrid Architecture for Learning Robot Control Tasks

Manfred Huber and Roderic A. Grupen
Department of Computer Science
University of Massachusetts
Amherst, MA 01003

Abstract

Autonomous robot systems operating in an uncertain environment have to be reactive and adaptive in order to cope with changing environment conditions and task requirements. To achieve this, the control architecture presented in this paper uses reinforcement learning on top of an abstract Discrete Event Dynamic System (DEDS) supervisor to learn to coordinate a set of continuous controllers in order to perform a given task. In addition to providing a base reactivity through the underlying stable and convergent control elements, the use of this hybrid control approach also allows the learning to be performed on an abstract system model which dramatically reduces the complexity of the learning problem. Furthermore, the DEDS formalism provides means of imposing safety constraints a priori, such that learning can be performed on-line in a single trial without the need for an outside teacher. To demonstrate the applicability of this approach, the architecture is used to learn a turning gait on a four-legged robot platform.

Introduction

Autonomous robot systems operating in an uncertain environment have to be able to cope with new situations and task requirements. Important properties of the control architecture of such systems are thus that it is reactive, allows for flexible responses to novel situations, and that it adapts to longer lasting changes in the environment or the task requirements. In many cases this adaptation is achieved using a learning process and has to occur without the direct influence of an outside teacher, making the reinforcement learning paradigm(Barto, Sutton, and Anderson 1983; Barto, Bradtke, and Singh 1993) an attractive option since it allows to learn from the system's interaction with the environment. However, while these exploration-based techniques have been applied to simple robot systems(Gullapalli 1992) and in simulation(Barto, Sutton, and Anderson 1983; Lin 1993; Crites and Barto 1995), the complexity of the action and state spaces of most robots renders these methods impracticable for on-line learning of continuous control strategies on such systems. To address this, as well as the lack of reactivity in the presence of novel situations, behavior-based techniques have been combined with such learning techniques(Maes and Brooks 1990; Mahadevan and Connell 1992; del R. Millán 1996). While this somewhat expands the scope of problems that can be addressed, the ad hoc character of the behaviors and the control mechanism still severely limits the scope of such systems. Furthermore, most such learning systems still do not provide a means for introducing a priori knowledge, thus permitting the occurrence of catastrophic failures which is often not permissible in real world systems which potentially have to learn new tasks in a single trial. To address these issues, the control architecture presented here uses a set of stable and convergent continuous controllers which are coordinated using a DEDS(Ramadge and Wonham 1989; Sobh *et al.* 1994) defined on an abstract, discrete state space. The corresponding supervisor, represented as a nondeterministic finite state automaton forms then the basis within which the given task is learned. The use of such a hybrid control architecture as a basis for a learning task promises to make it possible to address more complex tasks and platforms. Much of this promise stems from the fact that the learning component can treat the resulting system as an event driven system rather than a continuous (or clock driven) one, while the progression of the system in the underlying physical space is controlled locally by the continuous control elements. This dramatically reduces the set of points at which the learning agent has to consider a new action to the times when certain control or sensor events happen. While this allows for optimal decision points to be missed if the corresponding sensor signals lie outside the scope of the current set of control and sensor alternatives, it also leads to a focus of attention and can dramatically reduce the time required to learn a policy for the given task. To illustrate this and to demonstrate the applicability of the approach, it has been applied to a walking task on a four-legged walking robot.

The Control Architecture

The approach presented here uses a hybrid control structure as an interface between the physical world and the learning component. This structure effectively reduces the size of the state space used by the learning component and provides a means to guide the exploration and thus to influence the learning performance explicitly. In order to achieve this, the underlying continuous control elements used here are stable and convergent closed-loop control policies. This implies that they divide the underlying physical space into a set of stable regions within which they drive the robot system towards an attractor. This attractor, in turn, can be characterized abstractly by means of predicates indicating the achievement of the functional goals of the associated control policies. If these controllers are executed until convergence, the behavior of the system can largely be described by these attractors, which therefore allow to transform the underlying continuous space into a set of discrete system equilibria. Using the convergence of controllers as control events, the behavior of the system can thus be modeled approximately as a hybrid DEDS with a discrete state space corresponding to the convergence predicates of the controllers. The action dependent choice of the predicates should thereby ensure that the discrete space encompasses all tasks directly addressable by the underlying closed-loop controllers. This DEDS then forms the basic substrate for the reinforcement learning problem and, through the formal techniques available in the DEDS framework and local models of the behavior of the individual controllers, allows constraints to be imposed a priori in order to limit the set of admissible actions to safe and relevant control alternatives(Huber and Grupen 1996). Control alternatives available to the DEDS and learning systems are thereby the individual closed-loop controllers, as well as the hierarchical, concurrent activation of multiple of these controllers using the "subject to" ("$\triangleleft$") constraint. Similar to nullspace control for linear systems(Yoshikawa 1990) this constraint prioritizes the control actions such that a lower priority controller operates within an "extended nullspace" of the higher priority ones and can thus not counteract their progress. This in turn ensures that the stability and convergence properties of the original controllers are inherited by the composite controllers. Using this, the learning component uses Q-learning(Watkins 1989) to acquire a control policy which optimizes the given reinforcement while maintaining within the range of control actions allowed by the DEDS supervisor. At the same time the exploration process allows to build an improved abstract system model by estimating the transition probabilities within the DEDS model. This overall architecture is shown in Figure 1.

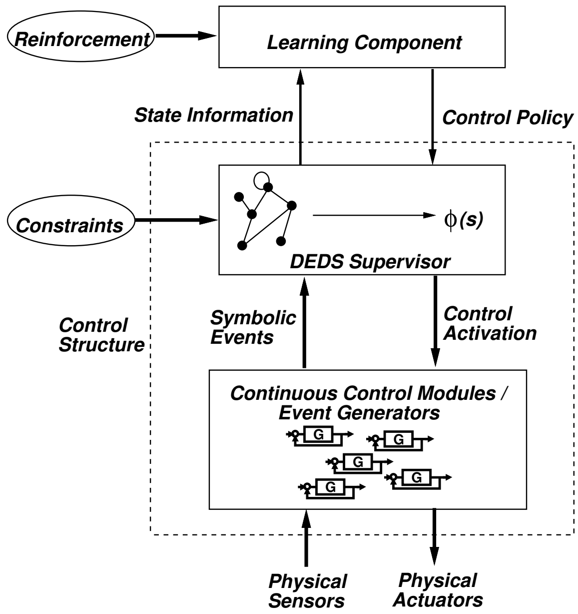

Figure 1: The Control Architecture

As shown in this figure, all direct sensory input and actuator output in this approach is handled by the continuous control elements in the bottom layer. Activation and convergence of these individual or composite controllers are then interpreted as discrete events in the abstract DEDS model of possible system behavior which forms the basis for the reinforcement learning system. A priori constraints imposed on this model can be used to limit the range of possible actions to keep the system within a safe mode of operation, as well as to implement temporally varying "maturational" constraints to improve learning performance. In addition to this, this structure also promises the possibility of hierarchical action spaces since learned control policies, together with the corresponding predicate space models, could be included as higher level controllers into the DEDS model and thus into the learning process.

Walking Experiment

To illustrate this hybrid learning and control architecture, the following shows an example of the overall architecture applied to the task of learning a turning gait on-line in a single trial on the four-legged walking robot shown in Figure 2.

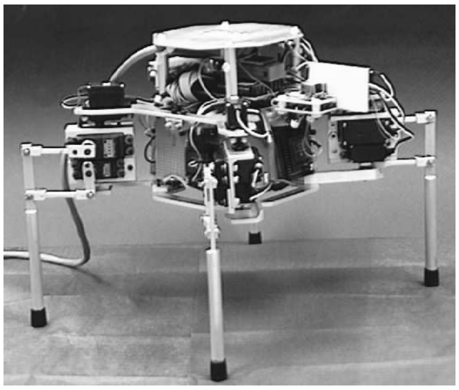

Figure 2: Walking Robot

Locomotion gaits are formed here as sequences of concurrent activations of a set of feedback controllers and represented as nondeterministic finite state machines. The set of feedback controllers used at the bottom layer of the architecture, is constructed using a control basis approach(Grupen *et al.* 1995). In this approach controllers are established by attaching a set of input resources (sensor abstractions) and output resources (abstract actuators) to a control law which addresses a generic control objective. In the case of the locomotion tasks, three control laws are used:

Φ_0: **Configuration space motion control -** a harmonic function path controller is used to generate collision-free motion of the robot in configuration space(Connolly and Grupen 1993).

Φ_1: **Contact configuration control -** contact controllers locally optimize the stability of the foot pattern based on the local terrain(Coelho Jr. and Grupen 1997).

Φ_2: **Kinematic conditioning control -** a kinematic conditioning controller locally optimizes the posture of the legs.

Each of these control laws Φ_i can be bound on-line to input resources σ and output resources τ derived as subsets of the system resources (legs $0, 1, 2, 3$ and position and orientation x, y, φ) of the four-legged robot illustrated in Figures 3.

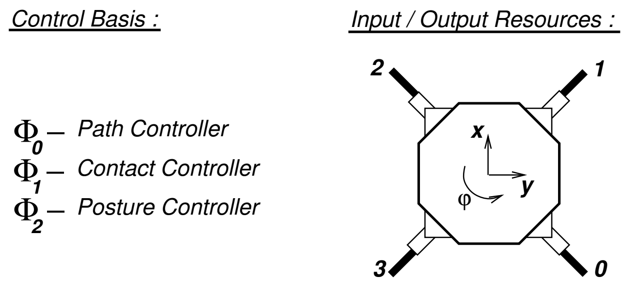

Figure 3: Controller and Resource Notation

The resulting feedback controllers $\Phi_i\frac{\sigma}{\underline{\tau}}$ can be activated concurrently under the "subject to" ("$\lhd$") constraint. The composite controller $\Phi_2\frac{0,1,2,3}{\underline{\varphi}} \lhd \Phi_1\frac{0,1,2}{\underline{0}}$, for example, attempts to achieve a stable stance on legs 0, 1, and 2 by moving leg 0 with the dominant controller while the subordinate controller optimizes the kinematic posture of all four legs within the "nullspace" of Φ_1 by rotating the body. For the example presented here, the set of possible controllers was limited in order to allow for a concise notation for the predicate space model. The set of closed-loop controllers available to the system consists here of all instances of the contact configuration controller of the form $\Phi_1\frac{a,b,c}{\underline{a}}$, where $a, b, c \in \{0, 1, 2, 3\}$, $a \neq b \neq c \neq a$ are three legs of the robot, and one instance of the kinematic conditioning controller, $\Phi_2\frac{0,1,2,3}{\underline{\varphi}}$. Using this set of 13 primitive controllers, the "$\lhd$" constraint can be used to construct a total of 157 composite controllers available to the DEDS and learning components. In addition, this choice of continuous controllers limits the set of convergence predicates to 5 elements $(p_1, p_2, p_3, p_4, p_5)$ since multiple controllers have identical control objectives and their predicates can thus be combined. The 5 predicates correspond to the convergence of individual controllers in the following way:

$$p_1 \leftarrow \Phi_1\frac{1,2,3}{\underline{*}}\ ,\ p_2 \leftarrow \Phi_1\frac{0,2,3}{\underline{*}}\ ,\ p_3 \leftarrow \Phi_1\frac{0,1,3}{\underline{*}}\ ,$$
$$p_4 \leftarrow \Phi_1\frac{0,1,2}{\underline{*}}\ ,\ p_5 \leftarrow \Phi_2\frac{0,1,2,3}{\underline{*}}\ ,$$

where $\underline{*}$ is a wildcard and indicates the independence of the predicate evaluation from the output resource. These predicates, together with initial, abstract models of the behavior of the controllers, form then the basis of the DEDS system which represents the space of all possible system behavior. The DEDS framework allows then to impose a quasistatic walking constraint of the form $p_1 \vee p_2 \vee p_3 \vee p_4$ (at least one stance has to be stable at all times) to determine the set of admissible control actions in each of the abstract predicate states.

To address a new task, Q-learning(Watkins 1989; Watkins and Dayan 1992), a widely used temporal difference method that learns control actions that maximize the expected future reward, is used to acquire a control policy for a given reinforcement signal on top of the constrained DEDS model. This scheme allows the acquisition of control policies even if their objective is not represented as a state in the underlying state space, and thus permits cyclic policies. In the experiment presented here an immediate reinforcement proportional to the rotational progress, $r_t = \varphi_t - \varphi_{t-1}$, is used to acquire a counterclockwise rotation gait. The safety constraint imposed in the DEDS layer allows thereby to simply start the robot in an arbitrary configuration on a flat surface and to learn the policy on-line in a single trial. A characteristic learning curve for this task is shown in Figure 4.

This graph, in which a control step indicates one controller activation, i.e. one transition in the DEDS model, shows that the robot rapidly acquires a good policy for this rotation task. The complete learning example executes on the real platform in approximately 11 minutes.

At the same time that such a policy is learned, the exploration can also be used to estimate transition probabilities between predicate states and thus to improve the abstract model of the system behavior. Such a model can be useful for off-line learning, as well as for system analysis and planning purposes in future tasks. Figure 5 shows the learned policy and the corresponding system model.

Here the numbers in the states represent the values of the 5 predicates, the controller definitions on the right indicate the learned policy for the core of the turning gait, and the width of the transition arrows indicates the acquired transition probabilities, with bold arrows

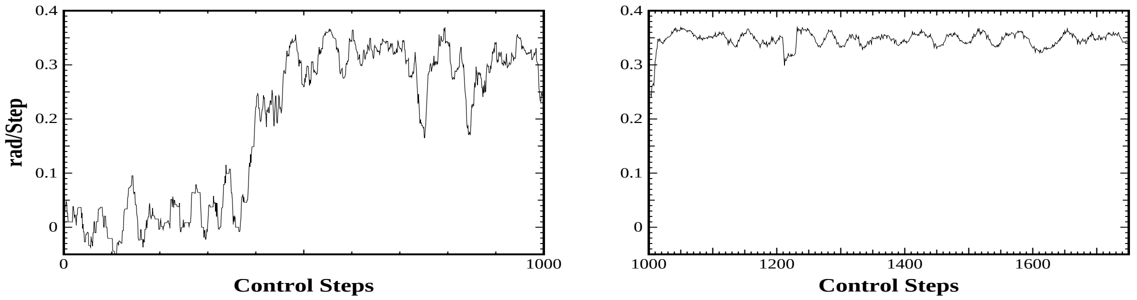

Figure 4: Learning Curve for Counterclockwise Rotation Task (left) and Performance of the Learned Policy without Exploration (right)

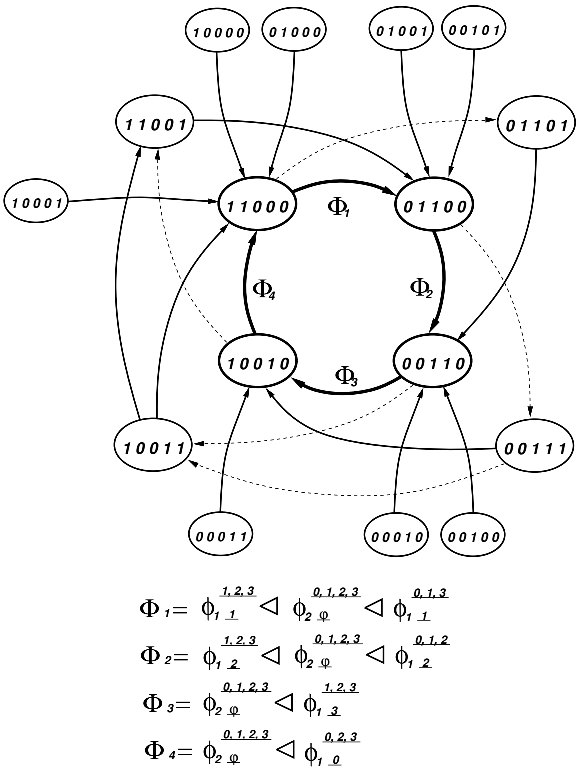

Figure 5: Learned Rotation Gait

for the central gait cycle indicating probabilities greater than 98%. The execution of this central cycle, which effectively leads the system through a sequence of stable, three-legged stances, is also depicted in Figure 6.

These robot pictures illustrate the angular progress achieved throughout execution of one gait cycle. The middle schematic, in which circles correspond to foot locations and the cross indicates the center of mass, shows the support polygons maintained throughout each controller transition, demonstrating that the system is always in a safe state throughout the execution of the learned policy.

Conclusions

The hybrid control architecture presented in this paper is designed to address on-line learning and control in complex systems and unstructured environments. To achieve this it employs a set of closed-loop controllers together with a DEDS layer which allows to incorporate certain types of a priori knowledge into the system and permits action dependent state abstractions in order to reduce the complexity of the subsequent learning problem. The learning example presented in this paper and other locomotion experiments performed using this architecture(Huber and Grupen 1998) show that this represents a feasible approach to perform learning for more complex tasks on-line on real robots. In addition, the use of such a hybrid control scheme allows to reason at a more abstract level within the discrete component of the architecture while the continuous aspects are locally controlled by the continuous control elements. This in turn promises to facilitate the design and construction of the control system, as well as to allow the use of a large variety of planning methods to aid in the construction of task specific control policies, and thus to further improve the adaptivity and autonomy of robot systems.

Acknowledgments

This work was supported by NSF IRI-9503687, IRI-9704530, and CDA-9703217.

References

Barto, A. G.; Bradtke, S. J.; and Singh, S. P. 1993. Learning to act using real-time dynamic programming. Technical Report 93-02, University of Massachusetts, Amherst, MA.

Barto, A. G.; Sutton, R. S.; and Anderson, C. 1983. Neuronlike adaptive elements that can solve difficult learning control problems. *IEEE Trans. Syst. Man Cyber.* 13(5):834–846.

Coelho Jr., J. A., and Grupen, R. A. 1997. A control basis for learning multifingered grasps. *J. Robotic Sys.* 14(7):545–557.

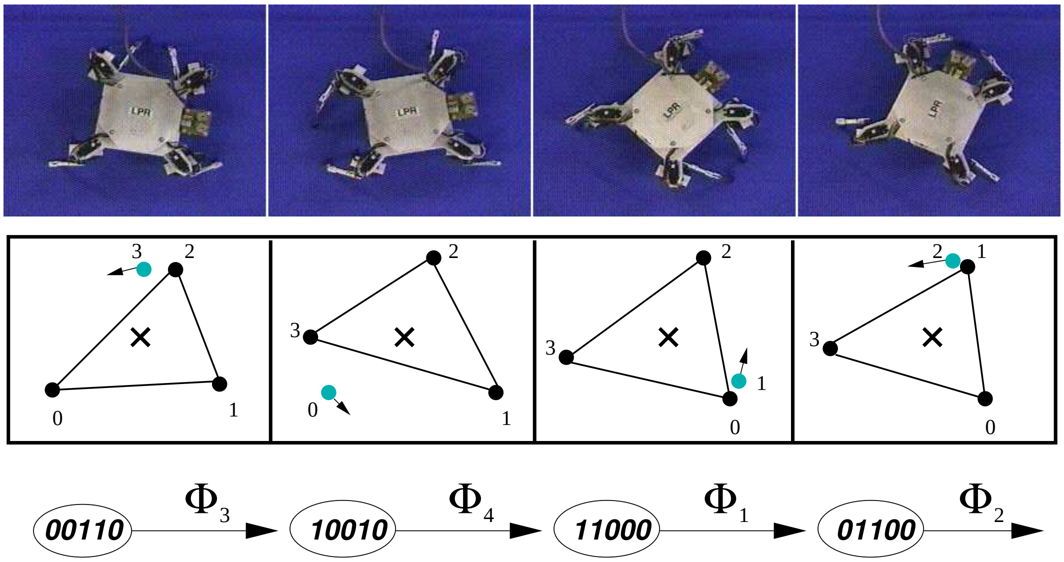

Figure 6: The Robot Executing the Central Gait Cycle of the Learned Policy (top), the Associated Stability Regions (middle), and the Corresponding Predicate State Transitions (bottom)

Connolly, C. I., and Grupen, R. A. 1993. The applications of harmonic functions to robotics. *J. Robotic Sys.* 10(7):931–946.

Crites, R. H., and Barto, A. G. 1995. Improving elevator performance using reinforcement learning. In *Advances in Neural Information Processing Systems 8.* Morgan Kaufmann.

del R. Millán, J. 1996. Rapid, safe, and incremental learning of navigation strategies. *IEEE Trans. Syst. Man Cyber.* 26(3):408–420.

Grupen, R. A.; Huber, M.; Coelho Jr., J. A.; and Souccar, K. 1995. Distributed control representation for manipulation tasks. *IEEE Expert* 10(2):9–14.

Gullapalli, V. 1992. Learning control under extreme uncertainty. In *Advances in Neural Information Processing Systems 5.* San Mateo, CA: Morgan Kaufmann.

Huber, M., and Grupen, R. A. 1996. A hybrid discrete event dynamic systems approach to robot control. Technical Report 96-43, CMPSCI Dept., Univ. of Mass., Amherst.

Huber, M., and Grupen, R. A. 1998. A control structure for learning locomotion gaits. In *Seventh International Symposium on Robotic and Applications.* Anchorage, AK: TSI Press.

Košecká, J., and Bogoni, L. 1994. Application of discrete event systems for modeling and controlling robotic agents. In *Proc. IEEE Int. Conf. Robot. Automat.*, 2557–2562. San Diego, CA: IEEE.

Lin, L.-J. 1993. *Reinforcement Learning for Robots Using Neural Networks.* Ph.D. Dissertation, Carnegie Mellon University, Pittsburgh, PA.

Maes, P., and Brooks, R. 1990. Learning to coordinate behaviors. In *Proceedings of the 1990 AAAI Conference on Artificial Intelligence.* AAAI.

Mahadevan, S., and Connell, J. 1992. Automatic programming of behavior-based robots using reinforcement learning. *Artificial Intelligence* 55:311–365.

Ramadge, P. J., and Wonham, W. M. 1989. The control of discrete event systems. *Proceedings of the IEEE* 77(1):81–97.

Sobh, M.; Owen, J.; Valvanis, K.; and Gracani, D. 1994. A subject-indexed bibliography of discrete event dynamic systems. *IEEE Robotics & Automation Magazine* 1(2):14–20.

Stiver, J.; Antsaklis, P.; and Lemmon, M. 1996. A logical approach to the design of hybrid systems. *Mathematical and Computer Modelling* 27(11/12):55–76.

Watkins, C., and Dayan, P. 1992. Technical note: Q-learning. *Machine Learning* 8:279–292.

Watkins, C. J. C. H. 1989. *Learning from Delayed Rewards.* Ph.D. Dissertation, Cambridge University, Cambridge, England.

Yoshikawa, T. 1990. *Foundations of Robotics : Analysis and Control.* Cambridge, MA: MIT Press.