# Contextual Control Policy Selection

**Jefferson A. Coelho Jr., Elizeth G. Araujo, Manfred Huber, Roderic A. Grupen**
Laboratory for Perceptual Robotics *
Department of Computer Science
University of Massachusetts
Amherst, MA, 01003 USA
{coelho, araujo, huber, grupen}@cs.umass.edu

## Abstract

Every autonomous agent operating in realistic settings must deal with incomplete state information. Sensory limitations, due to hardware constraints and/or limited interpretation algorithms, introduce hidden states which can prevent the acquisition of optimal control policies for a given task. This paper addresses this problem within the dynamical systems framework. The idea is to treat the agent in its environment as a dynamical system, and augment the original state space using contextual cues extracted empirically as the agent exercises existing control policies. Contextual cues are provided by the correlation between dynamic features of the agent-environment interaction and agent performance. In principle, the augmented state space allows the agent to learn to exploit control policies whenever they are likely to succeed, gather more information if the agent's context is ambiguous, and explore control alternatives in unfavorable contexts. Initial experiments involving an agent with impoverished sensing capabilities in a simulated, dynamic environment suggest that this approach might offer a relatively simple way to extract meaningful context information from dynamical systems, and to enhance the agent's performance.

## 1 Introduction

For an autonomous agent acting in the real world it is important to be able to react to changes in the environment and to learn behavioral strategies without depending on outside supervision. To achieve such flexibility, a number of systems employing reinforcement learning techniques have been implemented (Barto et al., 1983; Mahadevan and Connell, 1990; Lin, 1993). In these approaches the agent acquires a policy by interacting with the environment, while observing a simple reinforcement signal. While this seems well suited to autonomous agents, the complexity of the exploration-based learning techniques used often renders them impracticable for many on-line learning tasks. To address this complexity issue some interest has focused recently on hybrid or modular learning approaches. Here the objective is to learn the best composition for a set of *controllers* (Meila and Jordan, 1997; Araujo and Grupen, 1996; Matarić, 1997) rather than the best activation sequence for a set of low-level control actions. Using controllers or even complete policies as actions in the learning scheme leads to smaller state and action spaces, allowing for the solution of more complex tasks (Grupen et al., 1995; Huber and Grupen, 1997). Moreover, controllers can also provide the system with a basic level of performance by providing solutions to important subproblems.

A practical problem arises in real world tasks due to unobservable or incomplete state information, resulting in a non-Markovian system for which learning techniques such as Q-learning (Watkins, 1989) will generally not derive optimal policies. To address the problem of hidden state information several approaches have been proposed which either explicitly model the system as a partially observable Markov decision problem (POMDP) (Littman et al., 1995) or which attempt to augment the state information on-line by explicitly incorporating a sufficient amount of past information (McCallum, 1996b). Both approaches attempt to resolve the non-Markovian character throughout the learning process and to learn an optimal policy for the developing Markovian system. In general, however, this is a computationally hard task and may easily become intractable even for moderately complex problems.

This paper addresses learning in non-Markovian environments within the control composition framework. The framework proposed trades optimality for reduced learning complexity; in fact, the main objective here is to improve on the performance of the existing control policies, by augmenting the state

* http://piglet.cs.umass.edu:4321

information with task-relevant context information. The approach proposed will be referred to as Contextual Policy Selection (or CPS, for short). It consists of three steps, namely (1) on-line acquisition of a dynamic models database for each control policy, (2) construction of a Bayesian model of system performance, and (3) learning control composition on the augmented state space. The complete approach is described in Section 2.

Section 3 describes the task used to test our approach, including the original state representation, the baseline control policies and their derivation. Task-specific details for the implementation of our approach are discussed in Section 4. Section 5 presents experiments performed using this approach on the foraging task, and Section 6 concludes this paper.

## 2 Contextual Policy Selection

Many researchers, when confronted with the problem of synthesizing a complex control law, have proposed partitioning the system state space into regions of competence, and associating a local control law to each region (Michie and Chambers, 1968; Rizzi et al., 1992). The collective formed by the local control laws defines a global control law that may attend many user-specified requirements, such as global convergence, robustness with respect to noise or plant variations, and others.

Although this divide-and-conquer approach appears quite intuitive ("the right tool for the right job"), there are variations on it that also deserve consideration. For instance, instead of assigning control policies to fixed regions of the state space, one could consider applying a fixed control law for the whole state space and then identify those regions in which the control law is superior to control laws previously considered ("finding the job of each tool"). This is the premise behind the contextual policy selection approach: given a set of control policies, one can infer context from the dynamic behavior of the system. Under some conditions, the resulting control law will have better overall performance than any of the individual control laws.

Why is this framework relevant to systems with hidden states? First, the use of policies as the synthesis elements instead of primitive actions reduces the complexity of a generally hard learning problem. Second, pre-defined control policies allow the agent to experiment and discover important aspects of its interaction with the environment early on in the learning process. Third, it is easy to derive (suboptimal) control policies for agents acting on partially observable environments. Standard reinforcement learning techniques applied to partially observable environments often derive distinct, suboptimal control policies, due in part to unobservable state information. If so, we expect that the CPS framework will bring context to bear in the policy selection process. Fourth, sometimes a number of distinct control policies is actually required to solve a problem optimally. Consider the case in which an agent must grasp cylinders and cubes in the environment, but the identity of each object is unknown to it. It is definitely possible to derive an average control policy for grasping both objects, but the use of an specialized grasp policy for each object would be clearly advantageous.

The ideas presented here were inspired by the concept of control preimage, first introduced by (Lozano-Pérez et al., 1984), and then developed within the multifingered grasp problem domain (Coelho Jr., 1993; Coelho Jr. and Grupen, 1997). The next sections discuss the steps of the CPS approach.

### *2.1 Acquisition of the Dynamic Models Database*

This first step consists in executing each policy in the agent's repertoire of control policies for a number of trials and storing the resulting paths (sequence of sensed states) in memory. Many distinct paths will be generated, as well as a number of similar paths, due to the structure introduced by the underlying control policy $\pi_i$. Each path in state space can be labeled with an attribute vector, evaluated at the end of each trial. The elements of the attribute vector are performance measures relevant to some task. The resulting database will represent how well the agent expects to fare in the current context or state, given its control repertoire and intrinsic sensory and motor limitations.

The formal definition of the dynamic models acquisition task requires the definition of the termination condition for each trial, the number of trials required until some meaningful experience can be acquired, and the memory requirements for this task.

To address the trial termination issue, we will constrain the control policies to the subset of policies with absorbing states; that is, only those poli-

cies with finite duration will be considered. Control policies can be designed to be convergent, or to terminate as the task objective is achieved.

Many dynamical system techniques can be used to determine the number of trials required for the construction of a representative dynamical models database. One can evaluate how much novelty each trial is bringing to the database already in memory; for instance, novelty can be measured as how much of the agent's actions can be predicted with the current dynamic models.

Memory requirements will depend greatly on the complexity of the agent-environment dynamics, and not much can be said in general. However, as other researchers have noted earlier (McCallum, 1996a; Atkenson, 1991) the cost of memory is typically much smaller than the cost of running a robot; it makes sense to cache data corresponding to a number of trials, and use it to advantage later.

### *2.2 State Space Augmentation*

The classic approach for dealing with non-Markovian systems is to augment the notion of state to be the sequence of the last $n$ observations. The underlying assumption is that the non-Markovian system can be modelled by a $n$-order Markovian system. The determination of $n$ is generally done empirically; in some cases no matter how large $n$ is, this reduction from non-Markovian to Markovian system cannot be accomplished.

Instead of attempting to find a good value for $n$, we will take a more pragmatic approach, and augment the state space using a scalar measure of the agent performance. This performance measure can be evaluated by matching current observations to the agent's experience captured in the dynamic models database. The added complexity of the resulting augmented state corresponds to a extra scalar dimension, instead of a $n-1$ dimensional vector of previous state observations. We believe that there are many interesting and complex problems that fit the framework described.

### *2.3 Bayesian Models of System Performance*

Once the dynamic models are available, the agent can compute its expected performance $r$ conditioned to the probability that the sequence of $k$ state observations $x = \{x_1, \ldots, x_k\}$ comes from any model in the database:

$$r = E[R|X = x, \pi] = \sum_{i=1}^{n} r_i p(R = r_i | X = x, \pi), \quad (1)$$

where $\pi$ is the current control policy, and $n$ is the number of models in the model database. As more observations are available (i.e., as $k$ grows), the estimation of $r$ becomes closer to its true value (assuming the universe of models is representative of the agent-environment dynamics), and fewer terms are significant in the summation above. The explicit dependence on $\pi$ has important consequences: one cannot carry a sequence of observations across changes in the control policy. As soon as the agent decides to change its control policy, the vector of observations $x$ must be reinitialized. So at every moment the agent has the choice to keep executing the current policy and refine the estimation of its performance, or change policies and re-start the estimation process.

Notice that the problem above belong to the class of Bayesian classification problems (Hanson et al., 1990), in which the objective is to use the evidence available (state observations) to compute the degree of belief such evidence comes from one of several classes (models, in our case). The framework can in principle deal with noisy observations, the presence of outliers (i.e., state observations that do not come from any model in the database), and discrete and continuous state spaces.

### *2.4 Policy Selection with Augmented States*

Once the estimation of future performance is available, there are many ways of incorporating it to the agent's state description. The next problem the agent must solve is the tradeoff between improved state resolution (keeping the current control policy) and exploitation (changing current control policy), as mentioned earlier. The agent has no *a priori* elements to solve this tradeoff; it must exercise both options and find out which one is more advantageous in each context.

We propose to use reinforcement learning to derive an optimal switching policy for a given reward function as a function of the augmented state information. The action space consists of a set of control policies available to the agent, each of which offers a reasonable level of performance. The goal of the reiforcement problem is to discover the context that indicates which policy is most relevant from each augmented state.

## 3 The Foraging Task

The experimental task used throughout this paper is the foraging task described in (Araujo and Grupen, 1996) in which a dynamic learning agent with limited sensing capabilities has to acquire policies which allow it to capture prey in a dynamic world. This in turn requires strategies for finding, approaching, and attacking a prey under the time pressure introduced by a limited store of energy. To succeed at this task and to optimize performance, the agent has to solve a complex stochastic sequential decision task. The agent acquires a policy for the combination of primitive control elements taken from a set of controllers (Braitenberg, 1984). To evaluate the agent's performance at this foraging task, two types of environments are used which differ in their degree of complexity. While in both cases the environment is initialized randomly at the beginning of each trial, all prey in the first environment are stationary and do not react to the dynamics of the agent. In the second environment, different types of prey are present and each type exhibits a characteristic behavior that interacts with the learning agent's dynamics. In addition, different types of prey provide different amounts of reward.

Figure 1 shows the second environment where the learning agent is represented by a frog and the different species of prey indicate their distinctive behaviors. This figure presents part of the complex state of the overall system. To the agent, however, most of this information is not accessible due to its sensory limitations. In this foraging task the learning agent can only perceive the location and identity of the closest prey within a coarse spatial observer, that uses 3 distance levels and 2 angular zones (*center* and *off-center*)[1]. Also available are the presence of a second prey within range, and the agent's own fuel level discretized into 3 levels. Due to this limited sensory capability, the agent's representation of the world is generally quite ambiguous, and the environment behaves as a non-Markovian system.

Figure 2 shows two states that are indistinguishable for the agent. The left side of this figure shows the two partially observable world states while the right side represents the discretized perceptions available to the agent, i.e its current situation. In this example, the second prey in the top state is out of range and can thus not be perceived by the agent, leading to hidden state. The presence of such hidden state in a reinforcement learning system can severely hamper the performance of the agent. Furthermore, methods like Q-learning are not guaranteed to converge under these conditions (Watkins and Dayan, 1992). As expected, the use of standard reinforcement learning procedures on this foraging task resulted in a set of different policies with varying performance. Techniques such as shaping (Gullapalli and Barto, 1992), Q-learning with eligibility trace (Araujo and Grupen, 1996) can significantly improve the quality of solutions compared to standard Q-learning.

[1] The agent can not differentiate between left and right.

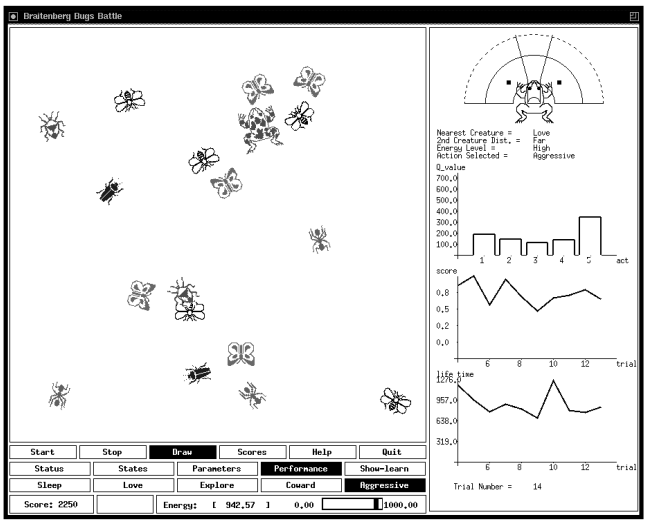


Figure 1 Simulated Foraging Environment

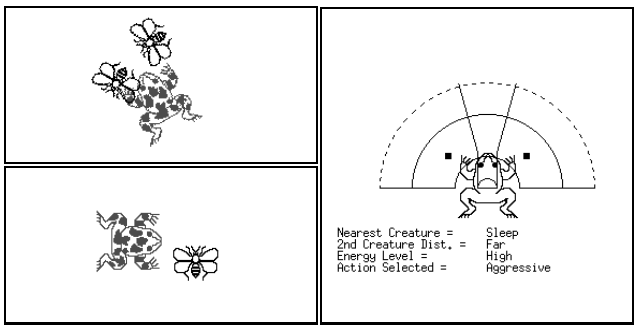


Figure 2 Hidden states

## 4 CPS and Food Foraging

The food foraging task just described is attractive for testing the ideas introduced in Section 2: it has many hidden states, a family of control policies with (possibly) distinct characteristics, and a well-defined performance measure.

However, in order to apply the CPS approach to this task, we segmented the trials into attacks, that

start and end when the frog adopts the "aggressive" behavior as it attempts to prey on some other bug near it. Each attack results in either success (the bug is eaten) or failure (poison bug is eaten or no bug is eaten). The dynamic models database for a policy $\pi_i$ is then built over a number of trials, while the agent follows $\pi_i$.

We experimented with two implementations of the performance estimation procedure; for the first implementation, the one step transition from the previous situation to the current situation is used as the estimate of the system dynamics. In this case, the dynamic models database can be simplified to a database of transitions and the respective probability that a bug is captured. We decided to build this database using 10 trials for the static environment (described in Section 3) and 30 trials for the dynamic environment. Figure 3 depicts the percentage of new transitions over the total number of transitions, as a function of the number of trials. The top curve corresponds to the dynamic environment, and the bottom curve corresponds to the static environment. New transitions are found mostly in the first trials, and the chance of finding new transitions decay with the number of trials. Figure 3 also shows that our dynamic models are not complete, if only the first 30 and 10 trials are used in their construction.

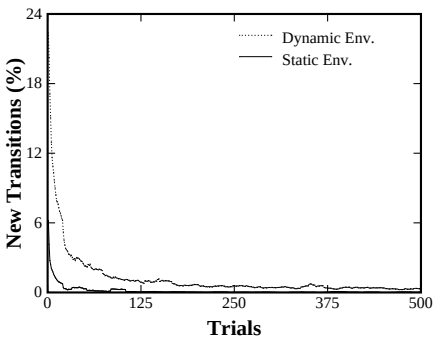


Figure 3 Percentage of new transitions found as a function of the number of trials. The top curve corresponds to the dynamic environment, while the bottom curve corresponds to the static environment.

The second implementation draws context information from the set of possible paths in which the system may find itself, given the sequence of sensed states. The dynamic models database was constructed based on data collected over 40 trials, for the static environment, and over 300 trials, for the dynamic one. The determination of the number of trials was done empirically, and it corresponds to the number of trials required before the percentage of new transitions taken during the last trial falls below 0.5%.

While executing a given policy, the performance estimation procedure relies on information extracted from the set of paths consistent with the observations made so far by the agent. As more transitions are performed, this set of paths typically will shed some of its elements and supply a more refined estimate. We are currently running experiments with this implementation, and the results are preliminary at this stage.

The augmented state representation is constructed by adding extra bits to the original state representation. The total number of extra bits will depend on the number of policies one wishes to compose, and on the discretization adopted for the percentage of good solutions ("successful attacks", for the foraging task) returned by the performance evaluation procedure. The results reported in Section 5 considered at most the composition of two policies, and only one level of discretization; therefore, only two extra bits were used to define the augmented state representation.

In the case of the transition-based evaluation procedure, the decision between "good" and "bad" is made based on the quality of the average incoming transition to the current state. For instance, in Figure 4, the average quality of incoming transitions to state $s_1$ is 20%. Therefore, transitions $t_1$ and $t_2$ would be flagged as good, since they are above average.

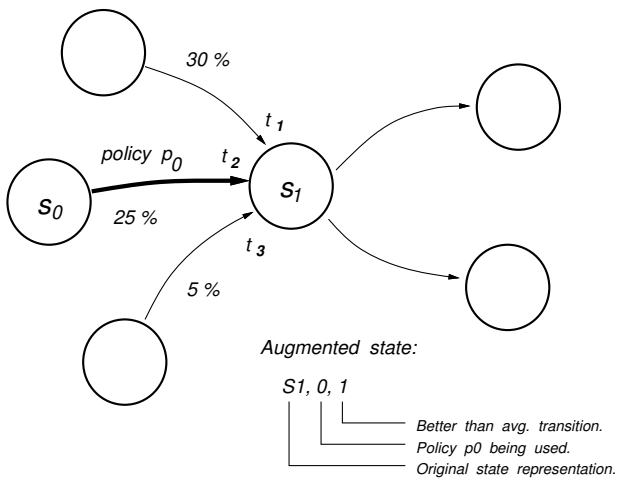


Figure 4 Augmented state information.

For the path-based evaluation procedure, the decision between "good" and "bad" prospects is based on the average percentage of successful attacks, as measured during the database construction stage. Given a set of paths, and respective success expectation, the system will regard the current context as good if the success expectation is above average.

## 5 Experiments

As mentioned earlier, experiments were performed in a static environment, where all prey is stationary, and in a dynamic environment, in which potential prey interact with the agent. In both cases, the agent learns about the utility of switching control policies using the Q-learning eligibility trace algorithm[2] with the following parameters:[3]

---

*Learning Parameters:*

- *learning rate* ($\beta = 0.025$);
- *discount factor* ($\gamma = 0.95$);
- *eligibility factor* ($\alpha = 0.95$);
- *initial temperature* = 1000.0 *(static environment),* 2000.0 *(dynamic environment);*
- *final temperature* = 1.0;
- *temperature decay factor* = 0.98;
- *Q-values were initialized with the Q-values of the corresponding policies, or with zero;*
- *the eligibility trace was initialized with zero;*
- *the state-action table is updated at each step of the simulation.*

---

The first experiment involved composing two policies for the static environment, one derived using Q-learning (P1) and the other derived with Q-learning eligibility trace (P2). Figure 5 depicts the performance histograms for both policies, computed over 300 test-trials (without exploration). P2 is a better policy, in terms of its mean performance (81% versus 57%) and its variance.

For this first experiment, the transition-based evaluation was used, and two configurations of the learning algorithm were tested; in the first one (M1(a)), the Q-table of the composition problem was initialized with the Q-values of the policies being composed. The second configuration (M1(b)) initialized the Q-table with zero. The left column of Figure 6 depicts the learning curves for both cases, preceded by a fragment of the learning curves for

[2] It is described in detail in (Araujo and Grupen, 1996).

[3] We performed little experimentation involving different learning parameters, due to time limitations.

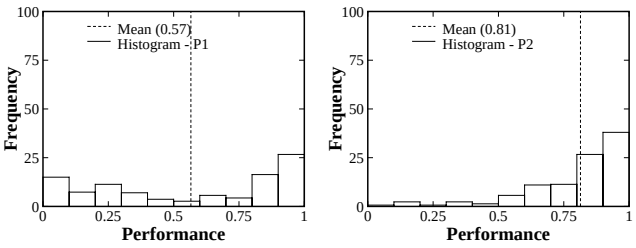


Figure 5 Performance histograms of policies generated with Q-learning (P1, left) and Q-learning eligibility trace (P2, right)

the policies being composed, for comparison purposes. Each point of the learning curve represents the average performance of 30 test-trials, and the error bars indicate the 95% confidence interval. A total of 950 learning trials were executed, and the histograms on the right correspond to the performance distribution over a total of 300 test-trials.

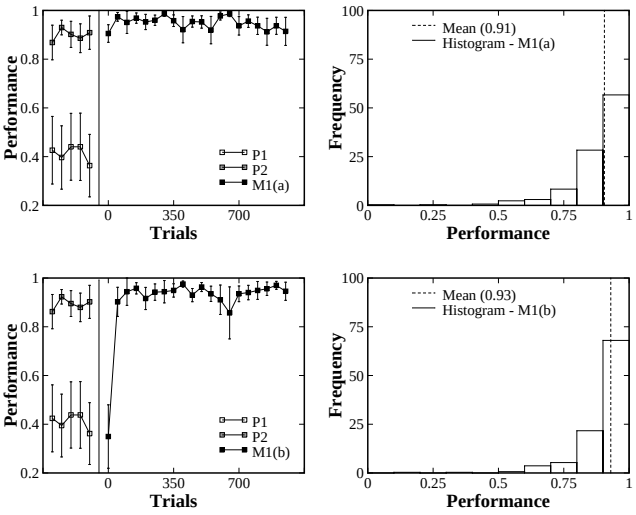


Figure 6 Learning curves and histograms for the policy composition using transition-based evaluation with Q-table initialized with the values from the corresponding policies (M1(a)), and Q-table initialized to zero (M1(b)).

As expected, M1(a) performed better initially, but the average performance after convergence, as evaluated by the additional 300 test-trials, is almost identical to the one of M1(b) (91% versus 93%, respectively). When compared to the best policy being composed (P2), the resulting policies are about 15% better in terms of average performance, and also exhibit a significantly smaller variance.

Figure 7 presents the learning curve for the composition of the same policies when and the path-based evaluation procedure is used to predict context information. When compared to the learning curves in Figure 6, the learning curve for this

case (M2) appears to be slower, more stable, with less peaks/valleys, and more consistent, as indicated by the smaller error bars. The average performance, however, is approximately the same as for the transition-based context evaluation procedure.

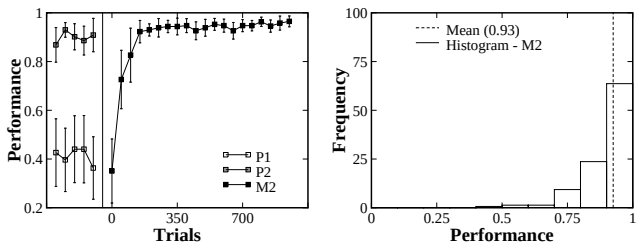


Figure 7 Results from the policy obtained using path-based evaluation procedure with the Q-table initialized with zero (M2).

The following experiment explored the benefits of composing policies for the dynamic environment. In this case, two policies constructed using the Q-learning eligibility trace algorithm were selected for composition; Figure 8 depicts the performance histograms for both policies (P3 and P4). As one can see from the average performance figures, the dynamic environment is a much harder learning problem.

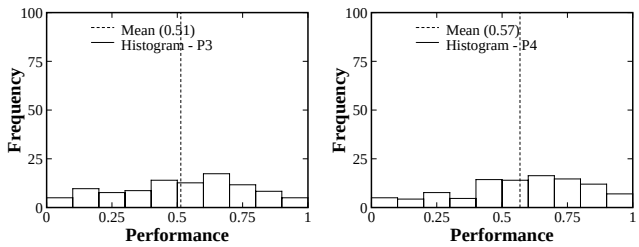


Figure 8 Performance histograms of two policies obtained with Q-learning eligibility trace (P3 and P4) in the dynamic environment.

Figure 9 shows the corresponding learning curve and performance histogram for the transition-based evaluation procedure (M4). As before, the average performance increased approximately 23%, from 57% to 70%, with a simultaneous decrease in variance.

The combined policy achieves the same performance level that was reached using handcrafted shaping procedures in the original reinforcement learning experiments. As opposed to shaping techniques, however, the composition approach presented here can achieve this performance autonomously. Furthermore, in an identical composition experiment using two policies learned in the

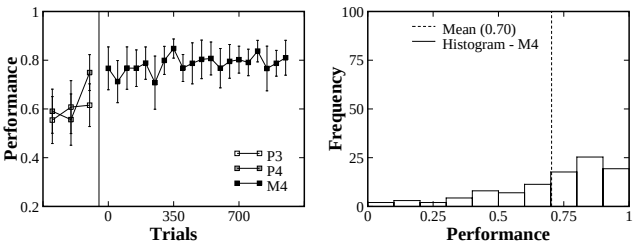


Figure 9 Results from the policy obtained with transition-based context evaluation procedure (M4).

dynamic environment using shaping, a composition policy was learned with an average performance of 80% which is comparable with the best handcrafted policies and outperforms all previously learned policies. These results suggest that the CPS approach presented in this paper might offer a relatively simple way to achieve higher performance solutions automatically from lower quality ones.

## 6 Conclusion and Future Work

The contextual policy selection approach maps system dynamics into performance expectations under an existing control policy. The resulting evaluation augments the available state information, and defines a search space in which existing control policies may be combined to improve overall performance. Our results showed that policy composition indeed performed better ($\approx 15\% - 23\%$), even though there was little room for improvement, given the quality of the policies being combined.

As for future work, we plan to evaluate how much contextual information the system is building as it learns to combine policies. In principle, the learning algorithm may improve its performance by not switching policies and building up context information until a more informed decision may be taken.

## Acknowledgments

The research reported here was conducted in the Laboratory for Perceptual Robotics at UMASS and supported in part by NSF under IRI-9503687, IRI-9704530, and CDA-9703217.

## References

Araujo, E. and Grupen, R. (1996). Learning control composition in a complex environment. In Maes, P., Mataric, M., Meyer, J.-A., Pollack, J., and Wilson, S. W., editors, *From Animals to Animats 4: Proceedings of the Fourth International Conference on Simulation of Adaptive Behavior*, Cape Cod, MA. MIT Press/Bradford Books.

Atkenson, C. (1991). Using locally weighted regression for robot learning. In *Proc. 1991 IEEE Int. Conf. Robotics*

*Automat.*, volume 2, pages 958–963, Sacramento, CA.

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.

Braitenberg, V. (1984). *Vehicles - Experiments in Synthetic Psychology*. M.I.T. Press, Cambridge, MA.

Coelho Jr., J. (1993). Effective multifingered grasp synthesis. Master's thesis, UMass, Department of Computer Science, Amherst, MA.

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

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

Gullapalli, V. and Barto, A. G. (1992). Shaping as a method for accelerating reinforcement learning. In *Proceedings of the 1992 IEEE International Symposium on Intelligent Control*, pages 554–559, Glasgow, Scotland, UK. IEEE.

Hanson, R., Stutz, J., and Cheeseman, P. (1990). Bayesian classification theory. Technical Report Technical Report FIA-90-12-7-01, Artificial Intelligence Research Branch, NASA Ames Research Center.

Huber, M. and Grupen, R. A. (1997). A feedback control structure for on-line learning tasks. *Robotics and Autonomous Systems*, 22(3-4):303–315.

Lin, L.-J. (1993). *Reinforcement Learning for Robots using Neural Networks*. PhD thesis, Carnegie Mellon University, Pittsburgh, PA.

Littman, M. L., Cassandra, A. R., and Kaelbling, L. P. (1995). Learning policies for partially observable environments: Scaling up. In *Twelfth International Conference on Machine Learning*.

Lozano-Pérez, T., Mason, M., and Taylor, R. (1984). Automatic synthesis of fine-motion strategies for robots. *The Int. J. Robotics Res.*, 3(1).

Mahadevan, S. and Connell, J. (1990). Automatic programming of behavior-based robots using reinforcement learning. Technical report, IBM Research Division, T. J. Watson Research Center, Yorktown Heights, NY 10598.

Matarić, M. J. (1997). Reinforcement learning in the multi-robot domain. *Autonomous Robots*, 4(1):73–83.

McCallum, A. K. (1996a). *Reinforcement Learning with Selective Perception and Hidden State*. PhD thesis, University of Rochester.

McCallum, R. A. (1996b). Hidden state and reinforcement learning with instance-based state identification. *IEEE Trans. Syst. Man Cyber.*, 26B(3):464–473.

Meila, M. and Jordan, M. I. (1997). Markov mixtures of experts. In Murray-Smith, R. and Johansen, T., editors, *Multiple Model Approaches to Modelling and Control*. Taylor and Francis.

Michie, D. and Chambers, R. (1968). Boxes: An experiment in adaptive control. In Dale, E. and Michie, D., editors, *Machine Intelligence 2*. Edingburgh.

Rizzi, A., Whitcomb, L., and Koditschek, D. (1992). Distributed real-time control of a spatial robot juggler. *IEEE Computer Magazine*, 25(5).

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*. PhD thesis, Cambridge University, Cambridge, England.