Dynamic Control Models as State Abstractions *

Jefferson A. Coelho Jr. and Roderic A. Grupen

Laboratory for Perceptual Robotics† - Department of Computer Science
University of Massachusetts, Amherst, MA, 01003 USA

ABSTRACT

This work proposes a methodology for the construction of state abstraction from a set of empirically derived models of system behavior. The idea is to treat the agent in its environment as a dynamical system and augment its observation space with *contextual cues* extracted empirically as the agent exercises each element out of the set of available control policies – the *control basis*. Contextual cues are provided by the correlation between dynamic features of the agent-environment interaction and agent performance. The resulting state abstraction (*observations* + *context information*) defines also a temporal abstraction, and offers interesting answers to some of the issues pertinent to the development of hierarchical systems. Initial experiments involving an agent with impoverished sensing capabilities in a simulated, dynamic environment demonstrate that relevant contextual information can be extracted and used to enhance the agent's performance.

1. INTRODUCTION

The most general computational framework for the study of autonomous systems is the POMDP framework, in which agents must deal with incomplete and uncertain observations, and unreliable control actions. Most of the problems studied with the POMDP framework can be characterized as *decision problems*, in which the optimal use of the information available is emphasized. System state is frequently expressed by a finite set of discrete values, derived from abstract sensors (e.g. existence of obstacles, doorways, etc.).

On the other hand, control theory researchers work on the development of algorithms and techniques applicable to a widening range of problems and circumstances. The control algorithms developed must deal explicitly with system dynamics and noise, expressed by continuous variables and probability distributions. Typically, optimal use of information (e.g. as in Kalman filtering) requires a model of the system dynamics and of the noise process, and it is not a goal in itself, but a means of achieving performance goals (e.g. tracking a target).

In this work, we propose to apply techniques originally developed for decision problems in the resolution of the *control composition* problem, where the agent is constrained to adopt one out of a set of pre-existing control policies. The methodology proposed consists of (1) modeling agent behavior through a finite set of stochastic observation models, as the agent adopts each control policy, and (2) the use of the set of observation models compatible with the data observed (or *context set*) as the agent's internal state.

The definition of the context set takes into account the agent's inherent computational and sensorimotor limitations, what guarantees a truly embodied representation of the agent experience. The re-description of the agent experience in term of context sets addresses the hidden state problem, as the context sets are formed taking into consideration all observations available. It also defines a model of temporal abstraction, where context changes are the segmenting events for the agent experience. The methodology proposed can be used by agents with both continuous and discrete observation spaces.

This work contributes to the discussion of abstraction and hierarchy in reinforcement learning by proposing

1. the use of controllers based on physical models of the interaction between agent and the environment as abstract actions; such formulation guarantees their transferability across tasks. Emphasis is put on (1) defining control policies with general applicability and (2) learning the utility of each control policy in the various contexts, rather than learning the control policies (or abstract actions) themselves.

2. the use of dynamic models of system behavior as the agent's internal state space. Ordinarily, sequence of observations are used as to construct state information; in the work proposed here, the existing controllers bias the sequence of observations, allowing the construction of the dynamic models that can be used in context identification. The representation chosen also addresses the hidden state problem, within the control composition framework.

3. the use of context change as the segmenting event of the agent experience. In term of temporal granularity, the proposed architecture is in between the proposed solutions of using clock ticks or termination predicates as segmenting events.

*This work was supported in part by NSF grants IRI-9503687 and IRI-9704530.

†http://www-robotics.cs.umass.edu

4. sequencing of controllers as the mechanism for scaling up the agent abilities.

The next section details the assumptions and the class of problems that can be addressed within the framework proposed. Section 3 presents the architecture in formal terms, followed by preliminary results (Section 4) and conclusion.

2. ASSUMPTIONS

The first assumption concerns the set of actions available to the agent: it consists of a set $\mathcal{A}$ of control laws, each one mapping observations to specific control actions. We have experimented with (1) feedback control laws, derived from locally accurate models of the interaction between the agent and the surrounding environment, (2) control policies learned through simplified observation models, as well as with (3) hard-coded rules relating observations to heuristic control actions. The control actions in $\mathcal{A}$ may be designed to address concerns such as safe operation of robots and faster learning. If feedback control laws are used, one can add noise rejection, and wide ranges of operation to the list of attractive features of this implementation. On the down side, it is obvious that the choice of control actions may limit the universe of tasks the agent may accomplish; in the general case, it is extremely hard to show the controllability[1] of a set of local controllers. We argue that for complex control problems, local optimization of control strategies may be the best one can hope for, at least in the near future.

Besides the availability of the *control basis* set $\mathcal{A}$, we will further assume that environment changes are significantly slow compared to the control rate associated with the elements of $\mathcal{A}$. In this situation, any perceived changes in the environment must be attributed to the actions induced by the elements of the control basis. In this quasistatic regime, the control actions structure the agent-environment interaction, at each step constraining the set of possible future observations. We will exploit this property in the definition of *context* below.

The third assumption concerns the existence of parametric models for the evolution of observation variables in phase space. In other words, it is assumed that the evolution of the observable variable ϵ can be captured by a stochastic model $M(\epsilon, \theta)$, where θ is a vector of parameters. Note that if ϵ is a quadratic function of the robot configuration, one can easily show that the associated gradient-based controller will always converge in the absence of noise. Furthermore, the corresponding set of observations $(\epsilon, \dot{\epsilon})$ can then be modeled by a locally linear model in phase space.

The fourth assumption establishes that the agent internal state can be represented by the set of models $M(\epsilon, \theta)$ compatible with the observations accumulated so far. As in the classic definition of state, the system "state" encapsulates all information needed to predict the behavior of the system. Therefore, knowing that a particular observation $(\epsilon, \dot{\epsilon})_i$ belongs to model $M_k(\epsilon, \theta)$ determines the state of the system; as in the original POMDP framework, no finite number of observations may determine the state of the system, as the domain of some models may overlap in phase space. The labels k of $M_k(\epsilon, \theta)$ define a discrete state space that can be used in lieu of continuous space variables.

[1] The ability to drive the system from any state to any other state, as defined in the context of linear control systems.

3. METHODOLOGY

The methodology consists of a collection of off-line and on-line procedures:

Behavior Sampling: Before dynamic models can be derived, one needs to sample the system behavior, as it is commanded by each control law in the set $\mathcal{A}$. The procedure consists of starting the system from random initial configurations, followed by the application of controller a_i until termination. All observations corresponding to each trial are stored in memory as a single unit; the trial data can be annotated with domain-specific performance indices or other auxiliary data.

A very important question to ask at this point is: how many trials are necessary before a significant sample of the system behavior is available? The general answer is open-ended, as we have not made any assumptions regarding the nature of the underlying dynamical systems. Nevertheless, one can borrow standard techniques from the dynamical systems literature for monitoring system behavior (i.e. detect phase transitions, proximity to singularities, etc) to estimate how much of the observation space has been visited. Indirect measures of how complete are the models acquired include the number of new models included in the last batch of trials, or the percentage of the agent's observations that can be explained by the existing models in memory. Our preliminary experience in the continuous robot control domain and the discrete foraging task domain show that it is possible to improve system performance even without complete models of the system dynamics.

Model Derivation: This off-line procedure computes the set of models corresponding to the data accumulated during the behavior sampling stage. The parameter vector θ that maximizes the log-likelihood of all observations in each trial dataset is computed, defining a model $M_i(\epsilon, \theta)$. Typically, there will be redundant models, or models with equivalent parameter vectors θ; such redundancy is reflected as the prior probability of the agent finding itself following that particular model.

State Inference: Given a set of stochastic observation models $\mathcal{M} = \{M(\epsilon, \theta_1), M(\epsilon, \theta_2), \ldots, M(\epsilon, \theta_n)\}$, and a sequence of observations it is a simpler matter to compute and update the belief vector that each model $M_k \in \mathcal{M}$ produced the observations. As opposed to the conventional Bayesian estimation rule, we propose two mechanisms for discarding certain models, namely by associating a domain to each model, as well as a bounded probability distribution to the observational noise process. Therefore observations outside model M_k's domain or with zero probability density will cause M_k to be ruled out as a plausible model for the observed dynamics. This simplification defines a partition in the set $\mathcal{M}$ between the models that could explain the data observed so far and the remainder. The first set is referred to as the *context set* K, and system behavior can now be described by transitions between context sets.

Whenever control laws, or macro actions are combined, one has to segment the agent experience into episodes during which the agent is not allowed to change its current policy. In one extreme case, the agent may allow changes at every servo cycle, or essentially at every clock tick; the other extreme is to run each policy until termination before evaluating the utility of adopting a different control policy. The description of the agent experience as a finite state machine where each state corresponds to distinct context sets introduce the possibility of using transitions between contexts as the segmenting event between episodes. In other words, transitions can be used as choice points for the agent to change its current policy, what is an attractive intermediate solution between the extremes described above. Note that changes in control policies reset the state inference procedure, as the policy itself is part of the definition of the system state.

Given n models, one can show that there can be $O(n^2)$ distinct context sets, under reasonable assumptions. Preliminary experimental results indicate that n^2 is a pessimistic upper bound on the total number of context sets (or total number of states).

The assumptions postulated in the previous section and the state inference procedure define all the elements of a reinforcement learning problem, that can then be solved using algorithms such as Q-learning or others:

- **S** (set of states): the set of all context sets K defined above;
- **A** (set of control actions): set of control policies, typically implemented as feedback control laws;
- **R** (reward function): defined by system designer.

Figure 1 depicts the elements of the architecture discussed above, in the context of a hypothetical problem involving the selection of two control policies π_1 and π_2. Its top row shows the phase plot diagrams for the control variables $(\epsilon, \dot{\epsilon})$ as policies π_1 and π_2 (left and right, respectively) are applied by the agent. Policy π_1 elicits 5 modes (A, B, C, D, E) of operation, represented by the respective models, while policy π_2 elicits modes (I, II, III, IV). Notice that the model domains overlap in phase space, as signaled by the gray areas in the figure. The bottom row of Figure 1 represents the corresponding state space. The transitions shown inside each oval will occur if the agent decides to follow the corresponding policy; in this case, the system will end up in one of the double-circled attractor states. In each of the states shown, the agent has the choice of changing policies from π_1 to π_2 and vice-versa.

4. PRELIMINARY RESULTS

We have applied the procedure described to the foraging task described in [Araujo and Grupen, 1996], where a dynamic learning agent with limited sensing capabilities has to acquire policies that 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. Adaptations were made in the basic algorithm described, as the observation space in this task is discrete.

The set of control actions consisted of two policies derived using Q-learning. Because of the hidden states, different runs of the learning algorithm result in distinct policies, each one a local minimum in policy space. To evaluate the learning procedure proposed here, two types of environments with distinct complexity were used. While in both cases the environment is initialized randomly at the beginning of each trial, all prey in the first environment (static 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 (dynamic environment). In addition, different types of prey provide different amounts of reward.

Figure 2 shows the second environment where the learning agent is represented by a frog and the different species of prey indicate their distinctive behaviors. The relative position of the bugs, their identity, and their velocity define the state of the environment. 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 neighborhood; this neighborhood is coarsely partitioned in 3 distance levels and 2 angular zones (*center* and *off-center*)[2]. Also available are the presence of a second prey within range, and the agent's own energy level discretized into 3 levels. Therefore, the observation space for static and dynamic cases consists of 36 and 180 possible observations, respectively. 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. Nevertheless, the data of only 40 (300) trials were used to build the set of system dynamics for the static (dynamic) case.

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

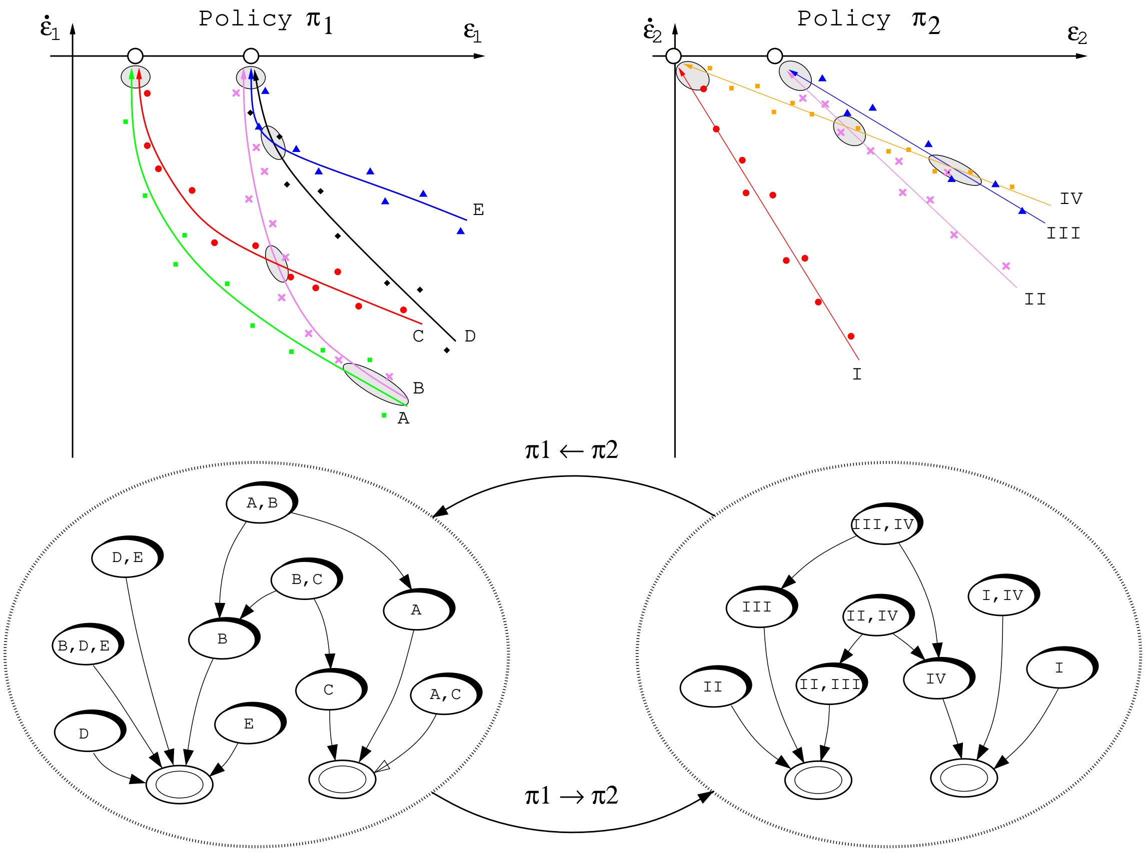

Figure 1: Top row diagrams depict the phase space plot $(\epsilon, \dot{\epsilon})$, for policy π_1 and π_2 (left and right, respectively). The gray areas represent ambiguous areas in two or more model domains overlap. The bottom row are the corresponding finite state diagrams.

4.1 Static Environment

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 3 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 (smaller) variance.

The left column of Figure 4 depicts the learning curve for the composite control policy (case M2), preceded by fragments of the learning curves for 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 histogram on the right correspond to the performance distribution over a total of 300 test-trials.

When compared to the best policy being composed (P2), the policy M1 is about 15% better in terms of average performance, and also exhibit a significantly smaller variance.

4.2 Dynamic Environment

In this case, two policies constructed using the Q-learning eligibility trace algorithm were selected for composition; Figure 5 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 6 shows the corresponding learning curve and performance histogram for the composite control policy (case M4). As before, the average performance increased approximately by 23%, from 57% to 70%, with a simultaneous decrease in variance.

The results obtained – a composite policy that is in average better than any of the policies being composed, and capable of reducing the variance associated with the agent performance – can be explained by the improved resolution of the agent's state space (making action selection more context-dependent).

4.3 Multifingered Grasp Domain

We have developed a tactile-feedback based grasp controller, capable of optimizing locally the grasp configuration over the surface of an object. Successive applications of the control algorithm cause the contacts to converge to a locally optimal grasp configuration. The control algorithm can be applied to any subset of the total number of grasping contacts (fingers), and distinct object geometries and/or task objectives may favor specific subsets of contacts – that is, the agent may benefit if the selection of which contact subset to use is made more context-dependent.

In the experimental setup considered, the agent may adopt

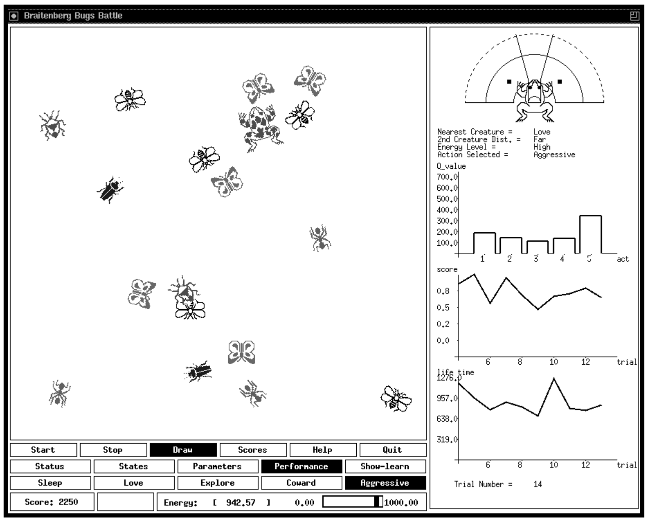

Figure 2: Simulated Foraging Environment

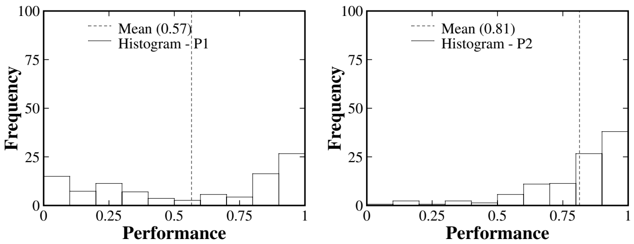

Figure 3: Performance histograms of policies generated with Q-learning (P1, left) and Q-learning eligibility trace (P2, right)

either a 2-contact or 3-contact grasp policy (the set $\mathcal{A}$), as it grasps one object out of a set of three objects (cube, cylinder, triangular prism). The reward associated with each trial is proportional to the number of probes required until convergence, plus the quality of the final grasp configuration, measured as the minimum friction required for stability.

The models of system behavior were constructed based on the data acquired over 120 grasp trials involving a set of three objects (cube, cylinder, triangular prism), for the 2- and 3-contact controller. A total number of 30 models have been constructed with the data gathered, resulting in 68 distinct con-

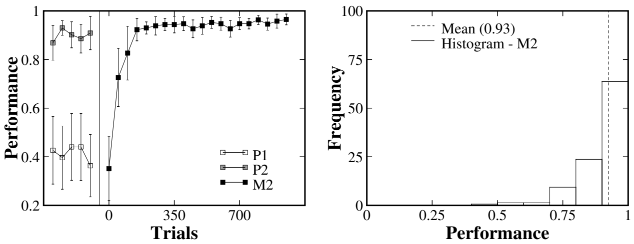

Figure 4: Learning curves and histograms for the composite control policy (M2).

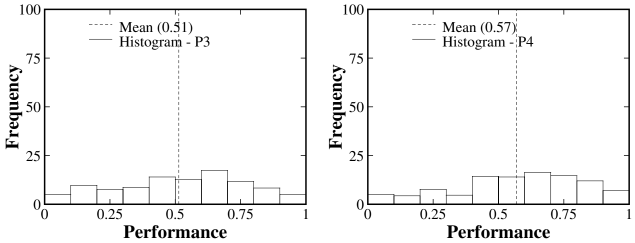

Figure 5: Performance histograms of two policies obtained with Q-learning eligibility trace (P3 and P4) in the dynamic environment.

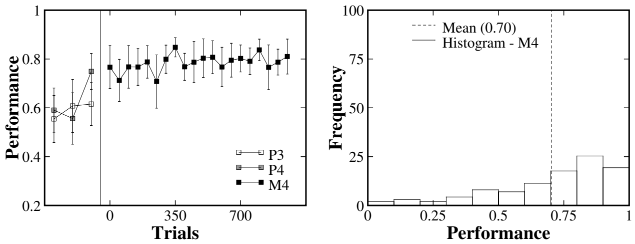

Figure 6: Results for the composite control policy (M4).

text sets (states) for the 2 contact controller, and 133 states for the 3 contact controller. We are in the process of implementing the learning procedure for determining the best policy to use in each context; we expect that object identity will strongly correlate with the choice of policy, even though such information (object identity) is not available to the agent. The experimental setup is described in detail in [Coelho Jr. et al., 1998].

5. Conclusion

The role of abstract actions in expanding the range of problems addressable by the reinforcement learning method has been acknowledged by many researchers. In this work, we propose an expansion of this role, by proposing that the dynamics elicited by certain abstract action can serve as an identifier for the current control context. We propose that much of the system behavior can be captured through a finite set of dynamic models; the models are by definition embodied and the corresponding universe of possible subsets (or *context sets*) defines a discrete, less ambiguous state space.

The description of the system state in terms of context sets allows the agent to experiment with combinations of the available control policies, and make the control selection procedure context-dependent. It also allows the use of context change events to segment the agent experience, a novel implementation of both state and temporal abstraction.

Note that the notion of "state" is not defined a priori, and therefore it is not observable by the agent; the interaction between agent and the environment provide the building blocks for the representation constructed. The framework is applicable to both discrete and continuous observation spaces, being especially relevant for implementation in physical robots.

References

[Araujo and Grupen, 1996] 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*, pages 333–342, Cape Cod, MA. MIT Press/Bradford Books.

[Coelho Jr. et al., 1998] Coelho Jr., J., Souccar, K., and Grupen, R. (1998). A control basis for haptically-guided grasping and manipulation. Technical Report CMPSCI Technical Report 98-46, Dept. Computer Science, University of Massachusetts.