Learning in Non-stationary Conditions: A Control Theoretic Approach

Jefferson Coelho COELHO@CS.UMASS.EDU
Rod Grupen GRUPEN@CS.UMASS.EDU
Laboratory for Perceptual Robotics, Dept. Computer Science, University of Massachusetts, Amherst, MA 01003

Abstract

Robots operating in dynamic environments must adapt their control strategy to each new situation presented; success is determined by the robot's ability in identifying transitions between distinct situations or contexts. This paper proposes a novel approach for partitioning the agent's experience into a set of discrete states. The state representation proposed is based on empirically-derived models describing the system dynamics induced by a set of control primitives. These generative models allow the agent to consider all possible outcomes and select the control primitive with highest utility in each context. The representation proposed is applied to the multi-fingered grasping problem, in which the agent must grasp objects with unknown geometries, independent of the initial relative orientation between hand and object. The results obtained show that indeed the representation proposed allows the robot to adapt its control strategy to each object (or context), even in the absence of specific sensors for object identity.

1. Introduction

The state of a system encompasses all information necessary to predict the future behavior of the system. Within robotics, state information is typically incomplete or partial, due to occlusions, limited sensors and computational resources.

Exact algorithms for the derivation of optimal control policies given partial state observability require the knowledge of observation and transition models, and are only applicable to systems with discrete state spaces (Kaelbling et al., 1998). Due to the inherent computational complexity, these algorithms are not of practical interest. Approximation algorithms typically monitor for violations of the Markov property, while searching for features in the past history of observations that complement the information currently available, in an attempt to build a state vector for the system. The search for predictive features is initially unbiased (e.g., as in the U-tree algorithm (McCallum, 1996)), as the agent explores the environment selecting random actions. Features are added to the state vector and gradually start influencing the agent's action selection; in fact, the relevancy of the first features may change due to the changes introduced in the agent's control policy by the newer features.

This general framework is inappropriate for tasks in the robotics domains; in these domains, biasing the search for relevant features is advantageous for two reasons: first, control laws based on the local models of how the agent affects its environment by changing its configuration are frequently available, and could be used to speed up the search for relevant features. Second, the controllers available can be designed to promote a safe exploration of the control alternatives.

Biped locomotion, obstacle negotiation, contact force modulation, and contact placement optimization are examples of tasks for which feedback controllers are available. The controllers must deal with distinct operational contexts, and incomplete state information. Terrain irregularities, obstacle configuration, contact friction and object identity affect the agent performance in these tasks, but are not part of the controllers' state vector due to the lack of specialized sensors. In spite of their shortcomings, these controllers are fairly competent once they are invoked in the right context; moreover, feedback controllers are robust with respect to variations in the plant they are controlling (e.g., object geometry, terrain properties).

In this work we propose a novel procedure for the construction of context-dependent controllers, based on the activation of the control primitive most appropriate to the perceived operational context or *state*. The control primitives are selected from a finite set of n controllers, referred to as the *control basis* Π. States are represented by the tuple $(\pi_i, \mathbf{q})$, where π_i is the control primitive currently active, and $\mathbf{q}$ is a boolean vector indicating which *generative models* are consistent with the last observation.

The effectiveness of the procedure is demonstrated within the multifingered grasp domain, where the grasping agent must reach a stable grasp configuration independently of the object identity and its relative orientation with respect to the manipulator. Distinct object identities and relative orientations define distinct operational contexts, but the corresponding information is unaccessible to the agent.

1.1 Related Work

One of the central premises of this work is the use of local controllers as the building blocks for a global control strategy. This idea has been explored by many researchers (Araujo & Grupen, 1996; Grupen et al., 1995; Huber & Grupen, 1997; Matarić, 1997; Michie & Chambers, 1968; Rizzi et al., 1992) when confronted with the problem of synthesizing a complex control law.

Reinforcement learning-based composition schemes (Araujo & Grupen, 1996; Huber & Grupen, 1997; Matarić, 1997; Meila & Jordan, 1997; Michie & Chambers, 1968) derive optimal control strategies through experimentation, and explicit models of system dynamics are not required; however, these methods require complete, accessible state information. The use of controllers or even complete policies as action primitives in learning schemes leads to smaller state and action spaces, allowing for the solution of more complex tasks (Araujo & Grupen, 1996; Huber & Grupen, 1997). It also allows for the construction of hierarchical solutions, involving control actions with distinct time scales (Sutton et al., 1999).

1.2 Dynamics Modeling

Dynamical models of system behavior are necessary for planning and inference; typically, they are not available, or are hard to derive. Most approaches equate system dynamics with the dynamics of sensory information. The Perceptual Distinctions Approach (PDA) (Chrisman, 1992) breaks away from the convention of estimating state information directly from percepts; instead, the agent experience is described in terms of transitions between internal states or *classes*. The classes are not directly observable, and the agent must maintain in memory a belief vector containing the probability that the agent is in each class.

The U-Tree algorithm (McCallum, 1996) records every (state, action, reinforcement) triple in memory, as observed by the agent. Each triple is also stored as a leaf in a *prefix tree*; each leaf may contain several triples. Leaf nodes can be split as relevant distinctions (i.e. distinctions leading to an improvement in the prediction of future rewards) are discovered, increasing the depth of the prefix tree incrementally. The leaf nodes are the states for the redefined reinforcement learning problem.

The U-Tree algorithm is still a very expensive algorithm, and simplifications are introduced in order to make it more manageable: first, there is a bound on the width of "time window" used to collect past observations/actions in order to build distinctions. Second, the order in which the distinctions can be performed must be predetermined by the system designer. Both simplifications may prevent the system from finding the optimal policy. However, McCallum's greatest concern is (in his words) the "chicken and egg problem": the current policy depends on the existing distinctions, and the distinctions formed depend on the current policy. Moreover, past distinctions may become irrelevant after changes in the control policy, and the U-Tree algorithm never removes distinctions proven to be relevant in the past – the U-Tree algorithm does not backtrack.

The algorithm proposed by Michaud and Matarić (1999) represent system state by the history of behavior activation, expressed in the form of trees. The algorithm exploits regularities in the environment (as captured by past experiences) in order to promote performance improvement.

We represent state using generative models of system dynamics, defined by the trajectory of controlled variables and their velocities in phase space. The resulting representation is more compact, and directly relevant to the objectives of the agent: there is no search for relevant features, as it is the case for the PDA architecture. It is also less explicitly related to raw perception than the representations employed by (Michaud & Matarić, 1999; McCallum, 1996), leading to better abstraction.

2. Derivation of State Information

The state representation proposed is based upon two discrete sets: the control basis $\Pi = \{\pi_1, \pi_2, \ldots, \pi_n\}$, a collection of feedback control primitives available to the agent for execution at each moment, and the membership pattern vector $\mathbf{q} = [p_1 \;\; p_2 \;\; \ldots \;\; p_m]^T$, that indicates which models out a set of m generative models are consistent ($p_k = 1$) with the last observation and which are not ($p_k = 0$).

Given the observation vector $\mathbf{o}$ collected as the agent executes the primitive π_i, the corresponding system state is denoted by the tuple $(\pi_i, \mathbf{q})$. The vector $\mathbf{q}$ expresses which models (or operational contexts) are compatible with the observation $\mathbf{o}$, according to the agent's past experience.

2.1 Control Basis

The control basis Π represents the agent's native control structure. Each $\pi_i \in \Pi$ promotes the optimization of a time-invariant, locally convex scalar field $\phi_i(x)$, by driving the robot configuration x along the gradient $-\nabla\phi_i(x)$. $\phi_i(x)$ measures the utility of the configuration x in robotic tasks such as for example navigation, posture, or force control. Typically, the definition of the controller $\pi_i(x) \propto -\nabla\phi_i(x)$ is analytic on the domain X (continuously differentiable and expressible as a power series for each $x \in X$), and it satisfies the Lipschitz condition (bounded gradient magnitude). The sum of these conditions defines an *autonomous*[1] dynamical system, characterized by static singular points. It follows that (1) the configuration space X of the robot can be partitioned into basins of attraction, regions defined by one attractor and all states that lead into it, when the system is driven by π_i; and (2) the underlying dynamical system is stable.

The stability of controller π_i ensures that the system will achieve a configuration x such that $\dot{x} < \delta$, where $\delta > 0$ is a threshold for equilibrium. The satisfaction of $\dot{x} < \delta$ determines the termination of π_i, segmenting the interaction between the robot and the environment into trials. The attractors satisfying the convergence predicate correspond to local optima of $\phi_i(x)$, due to the local nature of the underlying controllers.

The quality of the equilibrium configuration can be evaluated in several ways, as relevant to the task; the evaluation vector may include the number of control steps required until convergence, energy expenditure, or any scalar function $\mathcal{L}(x_f)$ of the equilibrium configuration x_f.

The definition of a control basis to address general problems is a hard task, as well as guaranteeing its completeness (i.e., that there always exists a sequence of controllers to address each problem instance). This is especially true for problems in which physical dynamic systems are not involved (e.g., pure discrete decision problems). Nevertheless, we claim that the control architecture presented in this paper addresses a rich problem domain, including the development of sensorimotor strategies. The architecture itself has been shown to be versatile and quite expressive (Grupen et al., 1995; Huber & Grupen, 1997).

[1] Autonomous dynamical systems are those systems that are not subject to any external influences that depend on time; $\pi_i(x) \propto -\nabla\phi_i(x)$ is a sufficient condition for autonomy.

2.2 Generative Models

A generative model describes how sequences of observations can be "generated" within a particular operational context. As the agent executes the controller π_i, it can find itself operating under many distinct conditions, characterized by distinct dynamics. If an exhaustive set of generative models is available, the agent may infer which models are compatible with a given sequence of observations by computing and updating the likelihood that each model has generated the data observed. If $P(M_k)$ is the likelihood that model k generates the sequence of observations, and $P(\mathbf{o}|M_k)$ the likelihood of observing $\mathbf{o}$ given that it is drawn from the distribution expressed by model k, then $P(M_k|\mathbf{o})$ (the likelihood that model k explains the data observed given observation $\mathbf{o}$) can be computed as

$$P(M_k|\mathbf{o}) = \frac{P(\mathbf{o}|M_k)P(M_k)}{\sum_{i=1}^{m} P(\mathbf{o}|M_i)P(M_i)},$$

assuming the existence of m generative models. The prior probability $P(M_k)$ can be determined using the data gathered during construction of the generative models. $P(M_k|\mathbf{o})$ can be interpreted as the belief that model k explains the observation $\mathbf{o}$.

The generative models compute the probability $P(\mathbf{o}|M_k)$; in this work, the corresponding conditional probability distributions will be represented by parametric models. The cost of acquiring data in real robots favors the choice of parametric models, which derivation typically require less data than for non-parametric ones. The use of parametric models is common practice in the dynamical systems literature (e.g., see Fraser and Dimitriadis (1993)), especially when insight about system dynamics is available.

2.2.1 Constructing Generative Models

The representation of the dynamics elicited by policy π_i requires a set of m generative models, expressed as $\mathcal{M}(\pi_i) = \bigcup_{k=1}^{m} M_{\pi_i}(\theta_k, \mathbf{o})$. The parameter vector θ_k describes the structure of the probability distribution $M_{\pi_i}(\theta_k, \mathbf{o}) = P(\mathbf{o}|M_k)$.

Each model $M_{\pi_i}(\theta_k, \mathbf{o}) \in \mathcal{M}(\pi_i)$ is constructed using data obtained empirically; as the agents executes the controller π_i, it records a sequence of n observations $\mathcal{O}=\{\mathbf{o}_1, \mathbf{o}_2, \ldots, \mathbf{o}_n\}$ gathered until convergence of π_i. The derivation of model $M_{\pi_i}(\theta_k, \mathbf{o})$ consists in determining the parameter vector θ_k^* that maximizes the likelihood that $M_{\pi_i}(\theta_k, \mathbf{o})$ "generates" $\mathcal{O}$.

The log-likelihood associated with sequence $\mathcal{O}$ is defined as $L(\mathcal{O}, \theta_k) = \sum_{j=1}^{n} \log M_{\pi_i}(\theta_k, \mathbf{o}_j)$, where n is the length of the sequence of observations $\mathcal{O}$, and $\mathbf{o}_j$ denotes the j^{th} element of $\mathcal{O}$. Many optimization procedures can be used to derive the parameter vector θ_k; in the special case in which $M_{\pi_i}(\theta_k, \mathbf{o})$ is a Gaussian distribution with an average that is a linear function of the parameter vector θ_k, there exists a closed-form solution for θ_k^*.

2.2.2 Dynamics Sampling

The construction of $\mathcal{M}(\pi_i)$ requires experience over a large set of control contexts. The problem of determining how much experience is necessary to construct meaningful models of system dynamics is an open one. A more complete set of models is certainly better for the agent's objectives, but it is hard to balance the utility of the missing information against the cost of acquiring it. Moreover, model completeness cannot be guaranteed, unless special assumptions are made. The issue of how much experience is necessary is related to the issue of exploration-exploitation in the reinforcement learning literature – see Thrun (1992) for a lucid review of the techniques proposed to address this issue.

The number of epochs required for sampling dynamics is an important parameter; it is an upper bound to total number of models in $\mathcal{M}(\pi_i)$. Its specification can be made within the framework of dynamical systems theory, where the traditional approach is to monitor how much "novelty" each interaction episode brings to the existing set of models. "Novelty" can be measured for instance by the percentage of the agent's time spent in unknown basins of attraction. If the exploration policy is assumed uniform, then model acquisition should stop as soon as the "novelty" index for the last n episodes falls below a given threshold τ.

Elimination of Redundant Models One generative model is derived for each sequence $\mathcal{O}$ recorded; after n trials, n models $M_{\pi_i}(\theta_k, \mathbf{o})$, $k = 1, \ldots, n$ will have been derived. Many of the n models derived represent a common dynamic pattern in phase space, and can be eliminated from $\mathcal{M}(\pi_i)$. Two models characterized by the parameter vectors θ_j and θ_k are said to be redundant if $|\theta_j - \theta_k| < \delta$; therefore, a clustering algorithm in parameter space Θ can be used to identify and purge redundant models. Frequency counts can be used to estimate the probability priors associated with each model in $\mathcal{M}(\pi_i)$.

2.3 Derivation of Membership Pattern q

The membership pattern $\mathbf{q} = [p_1 \;\; p_2 \;\; \ldots \;\; p_m]^T$ is a boolean vector containing m predicates, that indicate which models out of the set of m generative models are consistent with the last observation. The predicate p_k is defined as $p_k = (M_{\pi_i}(\theta_k, \mathbf{o}) > \kappa)$, where κ is an conveniently chosen threshold that reflects the confidence on the quality of the generative models.

As mentioned earlier, the state of the system is defined by the tuple $(\pi_i, \mathbf{q})$. Due to its definition, the resulting state representation is intrinsically situated, because it is grounded on autonomous interaction with the environment. Furthermore, it conveys more information than individual observations; the pattern $\mathbf{q}$ reflects the information derived from the agent's past experiences.

3. Multifingered Grasp Synthesis

Grasp synthesis requires the determination of the grasp configuration parameters (contact normals and relative contact positions with respect to the object's center of mass) that satisfy the minimum requirements set by the users in terms of a given grasp metric. Minimum friction coefficient required for stability, maximum admissible disturbance wrench, and the rank of the grasp matrix's null space are examples of grasp metrics.

While most researchers describe grasp synthesis as an optimization problem, we proposed that it is best characterized as a robust control problem. In this framework, the robot uses tactile feedback to compute incremental contact displacements, performed in order to optimize the grasp metric of interest.

For the experiments reported here, the contact displacements are determined by the grasp controller π_c described by Coelho Jr. and Grupen (1997). Contact displacements are computed based on local models of the interaction between the contacts and the object surface, and are aimed at reducing the squared wrench residual ϵ measured at the object's center of mass. The square wrench residual ϵ is defined as the sum of squared force and torque residuals. Given the wrench residual vector for n contacts

$$\rho = \sum_{i=1}^{n} \left[f_x^i \;\; f_y^i \;\; f_z^i \;\; \tau_x^i \;\; \tau_y^i \;\; \tau_z^i \right]^T,$$

then the squared wrench residual is defined by

$$\epsilon = \rho^T \rho. \tag{1}$$

The controller π_c displaces the subset c of contacts until a local minimum for ϵ is reached. The minimization of ϵ is closely connected to grasp stability: null wrench residual ($\epsilon = 0$) is a necessary condition for grasp stability (Coelho Jr. & Grupen, 1997), as it implies the existence of a null space of rank 1 or higher in the grasp matrix (Salisbury, 1982).

The subset of contacts c specifies which fingers and surfaces are enlisted in the grasp task. A hand with 3 fingers labeled $\{T, 1, 2\}$ allows for 4 distinct contact subsets, assuming that two or more fingers are required to grasp the object:

$$\mathcal{C} = \{(T,1),\;\; (T,2),\;\; (1,2),\;\; (T,1,2)\}. \tag{2}$$

Each instance of $c \in \mathcal{C}$ defines a new control law, affecting the outcome of the grasp process directly. The controller π_c is better characterized as an element of a family of grasp controllers, referred to as $\Pi = \{\pi_c |\; c \in \mathcal{C}\}$.

The control actions of π_c are independent of the global object geometry, depending solely on instantaneous, local tactile feedback. The convergent configurations for π_c correspond to local minima of ϵ, and for a given contact configuration, each choice of control law $\pi_c \in \Pi$ will lead to distinct convergent grasp configurations. Therefore, there exist an optimal choice in terms of which grasp resources c to use; this choice yields the convergent configuration with the minimum ϵ. The idea can be extended to controller sequencing: given a certain initial configuration, there exists an optimal sequence of controllers that lead the system state to the solution with the smallest possible ϵ. More importantly, controller sequencing expands the capabilities offered by individual controllers and allows one to build a system capable of adapting its strategy to the many operational contexts encountered. The construction of such context-dependent control strategy is described next.

4. Context-Dependent Grasp Policies

The problem to be addressed consists in selecting which sequence of grasp controllers one should employ to grasp an object, such that the coefficient of friction between fingers and object surface required to produce a stable grasp is minimized. The object's identity and its relative orientation with respect to the hand are unknown.

The determination of the optimal switching policy is a sequential decision problem, which can be solved within the Markov Decision Process (MDP) framework, provided one can describe the control switching as transitions between discrete states, within a Markovian state space. We assume that the state representation described in Section 2 satisfies the Markovian property, allowing us to use the Q-learning algorithm to approximate the optimal switching policy. This assumption guarantees the absence of limit cycles, a situation in which the agent selects a subset of the control primitives repeatedly. A description of the experimental setup and parameters follows.

4.1 Experimental Setup

The kinematics of the Stanford/JPL hand were used in the simulated grasp trials; it has three fingers, each one with three degrees of freedom. In each trial, the object to be grasped was selected from three possible object types: cylinders, cubes, and triangular prisms. The geometric parameters of the objects were drawn from a probability distribution, in an attempt to mimic measurement noise; as a result, no two grasp trials were performed on identical objects.

Initially, the hand was positioned next to the object top, in a configuration such that a patch of the object surface was within the workspace of each finger. This configuration favored top grasps, in which the fingers touch the vertical facets of the object surface. Figure 1 depicts the setup being used in pilot studies involving physical manipulators.

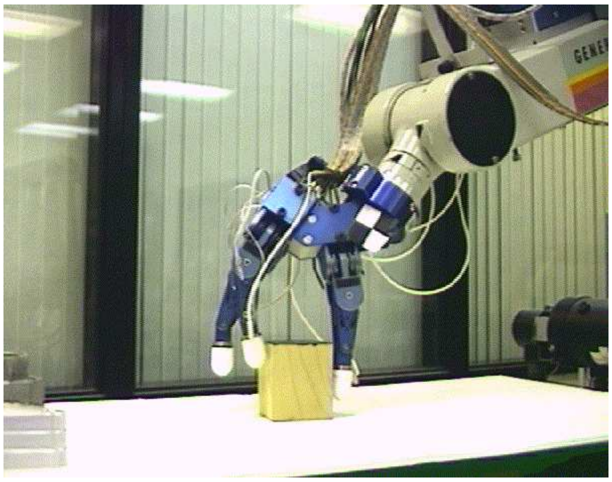

Figure 1. The setup for the grasp task: the hand is positioned over the object's top in a configuration that guarantees that all fingers reach the object.

4.2 Sequential Decision Problem

The state space representation proposed is based on a set of parametric generative models. Parametric models presume the existence of a generator mechanism for the data observed; the structure of the parametric model must be chosen according to the phenomenon one wants to model. For the multifingered grasp task, the following parameterization was used to represent the generative models:

$$\bar{\mathbf{o}} = [\epsilon \quad -K\epsilon + \epsilon_0]^T$$
$$M_{\pi_i}(\theta, \mathbf{o}) = \frac{1}{L}\frac{1}{\sqrt{2\pi\sigma^2}}\exp(\frac{-(\mathbf{o}-\bar{\mathbf{o}})^T(\mathbf{o}-\bar{\mathbf{o}})}{2\sigma^2}).$$

The observation vector $\mathbf{o} = [\epsilon \quad \dot{\epsilon}]^T$ consists of the squared wrench residual ϵ (defined by Equation 1) and its temporal rate of change $\dot{\epsilon}$. The parameter vector $\theta = [\sigma^2 \quad K \quad \epsilon_0]^T$ contains the parameters of the normal distribution, characterized by the variance σ^2, and the parameters K and ϵ_0 used to compute an estimate of $\dot{\epsilon}$. The constant L normalizes the distribution, such that

$$\int_{\mathbf{o}} M_{\pi_i}(\theta, \mathbf{o})\, d\mathbf{o} = 1.$$

The models $M_{\pi_i}(\theta, \mathbf{o})$ were constructed based on data collected over the course of 35 grasp trials for each family of objects. Considering the four possible choices for π_i (see Equation 2), a total of $4 * 3 * 35 = 420$ models resulted; after the elimination of the redundant models, 61 unique models were isolated.

Notice that the cardinality of the state space is given by the product $|\mathcal{C}| \times |\mathcal{Q}|$, where $\mathcal{Q}$ is the set of all distinct membership pattern vectors $\mathbf{q}$. It can be shown under reasonable assumptions that $|\mathcal{Q}|$ is proportional to the square of the number of models; in this case, the estimate corresponds to $\approx 61^2 = 3721$ states.

4.2.1 Reward Structure

No consensus exists with respect to which single grasp metric best captures the multiple dimensions along which a grasp configuration can be evaluated. In this work, the convergent grasp configurations received a score of $1 - \mu_0$, where μ_0 is the minimum friction coefficient required for grasp stability; a score of 1 ($\mu_0 = 0$) is best. No intermediate costs or rewards are present in the system.

4.2.2 Transition Semantics

The system evaluates the utility of pursuing a different control law whenever a change is detected in the membership pattern $\mathbf{q}$. Changes in $\mathbf{q}$ are special events, as they signal the fact that extra information has been acquired by the system.

After the convergence of the current control law, the system has the choice of terminating the trial or invoking a different controller. The insertion of a choice point after convergence of the current control law is necessary for two reasons: (1) it allows the system to use suboptimal convergent configurations as way points to the optimal configurations, and (2) the system's initial configuration may be at, or very close to, a suboptimal grasp configuration. The choice point allows the system to recover from these undesirable (albeit unlikely) situations.

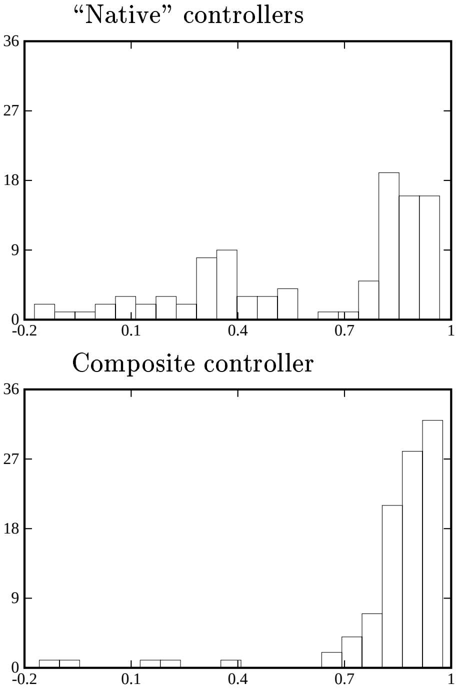

Figure 3. Performance histograms for the set of controllers in the control basis ("native controllers") and the composite controller, after learning. Histograms were built based on data collected over 100 trials. Grasp scores are represented in the horizontal axis; higher values are better.

5. Results

A total of 1600 learning trials were performed; in each trial an object type was randomly chosen, and an instance of it randomly generated. The initial control law was also chosen at random, determining which set of fingers probed the object. Up to 50 probes are employed in each grasp trial; after 50 probes, the trial is interrupted, and the current grasp configuration scored as if it were a convergent configuration. Exploration was regulated by a Boltzmann distribution.

Figure 2 depicts a typical learning curve (curve labeled △□○, top left curve). The curve was smoothed with a sliding window 10 data points wide. Each point cor-

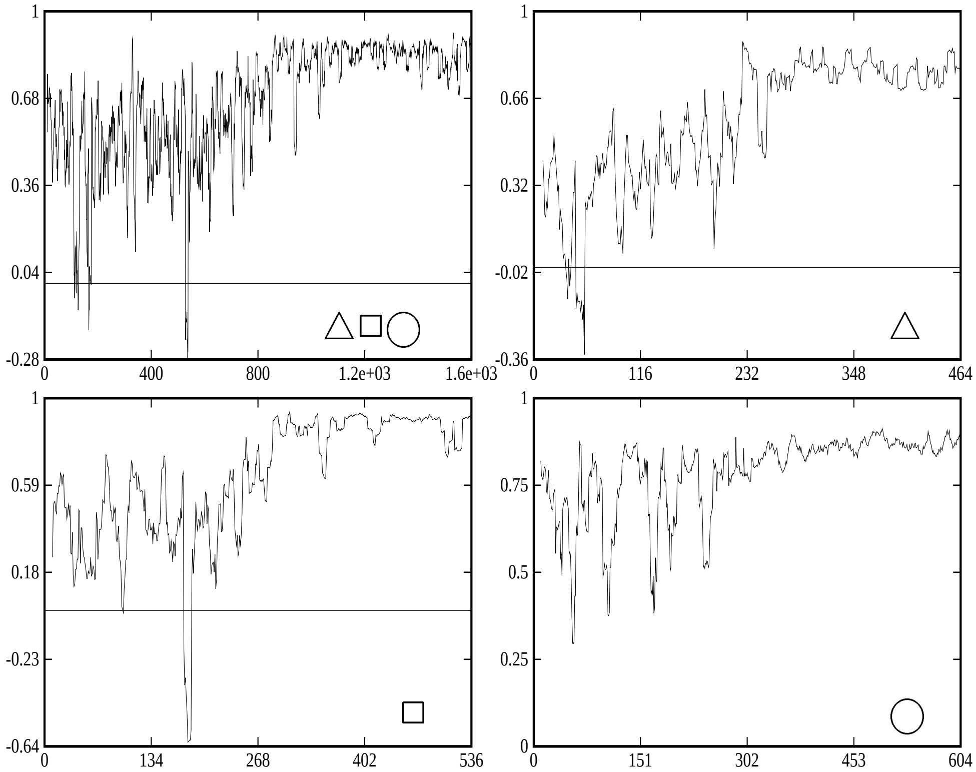

Figure 2. △□○ Typical learning curve for all objects. Other curves (labeled △, □, ○) are the corresponding learning curves for the individual objects. Vertical axes represent grasp scores $(1-\mu_0)$, and horizontal axes are the trial number.

responds to the grasp score associated with the convergent grasp configuration, for the randomly chosen object. The data corresponding to each object can be separated in three classes, corresponding to the three families of objects. The resulting learning curves are labeled △, □, ○.

It is worthwhile to mention that because the objects are not perfectly symmetrical, and the control execution is halted before $\dot{\epsilon}=0$, it is unrealistic to expect an average score of 1. The curves for the individual objects are close to the optimal, within the limitations of the Q-learning algorithm.

Figure 3 presents the performance histograms for the controllers in the control basis ("native" controllers), and for the composite controller after learning. Each histogram depicts data collected over 100 trials, involving all objects. For the top histogram, a controller was selected from the control basis at random, and executed until convergence. The convergent grasp configuration was scored and included in the histogram.

As one can see, the composite controller avoids the majority of low quality solutions: 93% of the solutions have scores higher than 0.7, compared to 56% for the native controllers. The variance associated with solution quality is also substantially smaller.

Table 1. Breakdown of controller invocations (leftmost column), as a function of object type.

	△	□	○
$(T,1)$	12.4 %	47.6 %	30.4 %
$(T,2)$	12.9 %	34.8 %	18.1 %
$(1,2)$	17.4 %	7.8 %	24.2 %
$(T,1,2)$	57.2 %	9.8 %	27.4 %

Table 1 reports how frequently each control primitive was invoked, as a function of object type. Notice that all controllers are used in all objects, an indication that control actions may be sometimes selected with the goal of information gathering and ambiguity resolution.

6. Conclusion and Future Work

Systems operating in non stationary conditions are required to identify their current operational context, and select the appropriate control strategy on the fly. We have described a solution in which system state is derived based on a mixture of empirically derived generative models. The method proposed was successfully applied to the multifingered grasp task, in which a robotic hand must deploy its grasp resources (fingers) according to the object identity and orientation, both unknown to the system.

In solving this task, the system develops its own notion of "object identity", and uses it effectively. Notice that object classification is not an explicit goal of the system; object categorization is only important insofar as it influences grasp performance.

We plan to implement the algorithm described in the LPR's arm and hand system, consisting of a General Electric P50 robot and a Salisbury hand. The hand is equipped with tactile sensors, capable of reporting contact position and normals with respect to a given point within the object. The number of trials is an issue, due to the cost of the probing operation. Like other practitioners (Lin, 1993; McCallum, 1996), we plan to store the data acquired in memory and re-use it to speed up learning.

Acknowledgments

This work was supported in part by the National Science Foundation under grants CISE/CDA-9703217, IRI-9704530 and IRI-9503687.

References

Araujo, E., & Grupen, R. (1996). Learning control composition in a complex environment. *From Animals to Animats 4: Proceedings of the Fourth International Conference on Simulation of Adaptive Behavior, 333–342*. Cape Cod, MA: MIT Press/Bradford Books.

Chrisman, L. (1992). Reinforcement learning with perceptual aliasing: The perceptual distinctions approach. *Proceedings of the Tenth National Conference on Artificial Intelligence*. Cambridge, MA: AAAI/MIT Press.

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

Fraser, A., & Dimitriadis, A. (1993). Forecasting probability densities by using hidden markov models with mixed states. In A. Weigend and N. Gershenfeld (Eds.), *Time series prediction: Forecasting the future and understanding the past*, 265–281. Santa Fe Institute/Addison-Wesley.

Grupen, R., Huber, M., Coelho Jr., J. A., & Souccar, K. (1995). Distributed control representation for manipulation tasks. *IEEE Expert*, *10*.

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

Kaelbling, L., Littman, M., & Cassandra, A. (1998). Planning and acting in partially observable stochastic domains. *Artificial Intelligence*, *101*.

Lin, L.-J. (1993). *Reinforcement learning for robots using neural networks*. Doctoral dissertation, School of Computer Science, Carnegie Mellon University, Pittsburgh, PA.

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

McCallum, A. K. (1996). *Reinforcement learning with selective perception and hidden state*. Doctoral dissertation, Department of Computer Science, University of Rochester, Rochester, NY.

Meila, M., & Jordan, M. I. (1997). Markov mixtures of experts. In R. Murray-Smith and T. Johansen (Eds.), *Multiple model approaches to modelling and control*. Taylor and Francis.

Michaud, F., & Matarić, M. (1999). Representation of behavioral history for learning in nonstationary conditions. *Robotics and Autonomous Systems*, *29*, 187–200.

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

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

Salisbury, J. K. (1982). *Kinematic and force analysis of articulated hands*. Doctoral dissertation, Department of Mechanical Engineering, Stanford University, Stanford, CA.

Sutton, R., Precup, D., & Singh, S. (1999). Between mdps and semi-mdps: A framework for temporal abstraction in reinforcement learning. *Artificial Intelligence*, *112*, 181–211.

Thrun, S. (1992). The role of exploration in learning control. In D. White and D. Sofge (Eds.), *Handbook of intelligent control: Neural fuzzy, and adaptive approaches*, 527–559. New York: Van Nostrand Reinhold.