Proceedings of the 2007 IEEE Symposium on Approximate
Dynamic Programming and Reinforcement Learning (ADPRL 2007)

Identifying trajectory classes in dynamic tasks

Stuart O. Anderson
Robotics Institute
Carnegie Mellon University
5000 Forbes Ave
Pittsburgh, Pennsylvania 15213
Email: soa@ri.cmu.edu

Siddhartha S. Srinivasa
Intel Research Pittsburgh
Intel Corporation
4720 Forbes Ave
Pittsburgh, Pennsylvania 15213
Email: siddhartha.srinivasa@intel.com

Abstract— **Using domain knowledge to decompose difficult control problems is a widely used technique in robotics. Previous work has automated the process of identifying some qualitative behaviors of a system, finding a decomposition of the system based on that behavior, and constructing a control policy based on that decomposition. We introduce a novel method for automatically finding decompositions of a task based on observing the behavior of a preexisting controller. Unlike previous work, these decompositions define reparameterizations of the state space that can permit simplified control of the system.**

I. Introduction

One common approach to finding control policies for robotic systems is to decompose a task into subtasks that are individually less difficult to solve. Methods like state space funnels [1], [7], behavioral primitives [2]–[4], hybrid systems [5], and reinforcement learning [6] are all variations of this approach. In practice, however, finding a useful decomposition is most often not automatic and relies on expert human knowledge of a particular problem domain.

Methods for automatically identifying equivalence classes of trajectories using geometric structure in phase space have been shown [8], [9]. It has also been demonstrated that, once identified, these structures can be used to generate controllers autonomously [10]. Recent work in the POMDP dynamic programming community [11], [12] has emphasized the value of observing existing control policies to guide the search for optimal policies. The key idea in this work is that an existing policy is a task independent representation of information about the task domain in which it operates. Our approach combines automatic problem decomposition with the use of information from observed policies. Rather than finding a decomposition to aid the search for a policy, we find a natural decomposition for a problem by observing a preexisting solution.

The method described in this paper decomposes observed motions of a system into exemplar motions based on a measure of the local rate of convergence. Each motion class has an associated exemplar motion that is representative of all the motions in that class. Exemplar motions are automatically generated and are used as the basis for building new parameterizations of state space that are unique to each motion class. We believe that it is often the case that the behavior of the observed policy can be closely reproduced by a linear controller operating in the reparameterized state space near an exemplar motion.

Section II of this paper presents definitions of the exemplar motions of a system and defines a reparameterization of state space based on these classes. Section III presents a method for numerically computing these motions and associated reparameterizations. Section IV presents an example of this method applied single link inverted pendulum swing up task. Section V discusses operation of the method in higher dimensions, provides theoretical justification for the methods described in section II, and compares this work to previous efforts. Section VI discusses ongoing and future work on extending the technique to deal with the sparseness of human motion capture data.

II. Definitions

This section presents a definition of exemplar motions based on observed trajectories as well as a reparameterization of state space that can simplify the control policy associated with each exemplar. Exemplar motions are found by searching for regions of state space where the flow of the system is strongly convergent and identifying trajectories about which the system is most convergent. A single stable region may have several exemplar motions if the rate of convergence varies significantly over the region. Because the definition of the exemplar motions is sensitive to variations in the rate of convergence, is is often the case that the rate of convergence toward an individual exemplar is fit well with simple parametric models. This can permit the use of linear control policies in the reparameterized state space. Fig.1 illustrates the definitions provided in this section.

Exemplar motions

Consider a time invariant, fully observable, deterministic system with state $\mathbf{x}$, control input $\mathbf{u}$, system dynamics $\dot{\mathbf{x}} = f(\mathbf{x},\mathbf{u})$ and control policy $\mathbf{u}^* = c(\mathbf{x})$. If $\mathbf{u}^*$ is used to control the system, then every point in the state space lies on some trajectory

$$G(d,\mathbf{x}) = \mathbf{x} + \int_0^d f(\mathbf{x},\mathbf{u}^*)\,ds \qquad (1)$$

Samples of these trajectories for the example problem discussed in section IV are illustrated in blue in Fig.4 and Fig.6.

1-4244-0706-0/07/$20.00 ©2007 IEEE

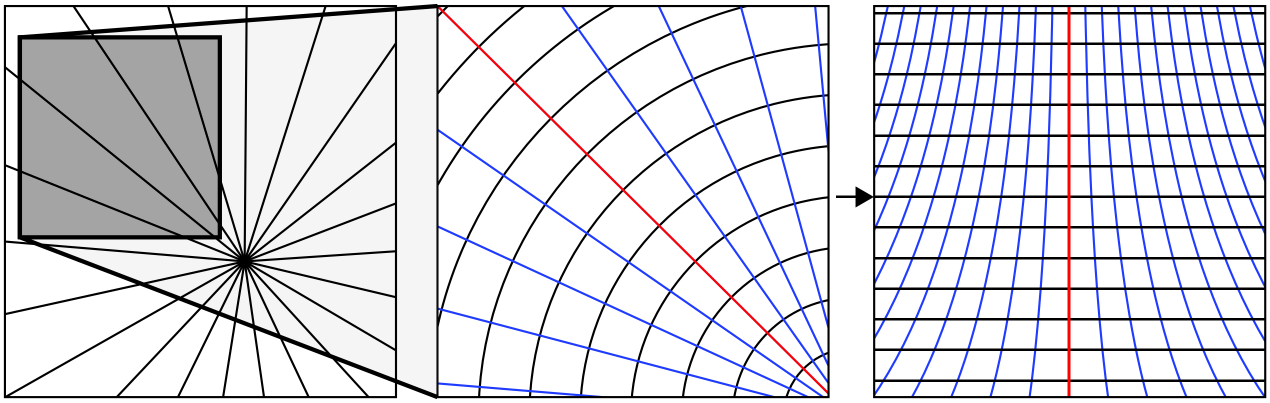

Fig. 1. Trajectories of a possible system(left), Tangent manifolds of a section of that system (center), and ST-Space reparameterization about a key trajectory of the system (right)

Qualitatively, exemplar motions of the system are the set of trajectories that other trajectories converge toward. Flow convergence is not an appropriate metric for defining these exemplar motions because we are not concerned with convergence in the direction of flow caused by a change in the rate of flow. However, a convergence metric that reflects only convergence tangent to the direction of flow is useful. Exemplar motions for the example problem are illustrated by red lines in in Fig.5 and Fig.6.

A ‘flow tangent manifold’ $M(\mathbf{x},d)$ is defined such that $f(\mathbf{x},\mathbf{u}^*)$ is normal to $M(\mathbf{x},d)$ at all points in $M(\mathbf{x},d)$ and $G(d,\mathbf{x}) \in M(\mathbf{x},d)$. Some flow tangent manifolds for the example problem are shown in black in figures Fig.5 and Fig.6.

Define the flow tangent divergence of $f(\mathbf{x},\mathbf{u}^*)$ to be the scalar quantity

$$q(\mathbf{x}) = \nabla \cdot (v(\mathbf{x},\mathbf{u}^*) f(\mathbf{x},\mathbf{u}^*)) \qquad (2)$$

where $v(\mathbf{x},\mathbf{u}^*)$ is an orthonormal basis of the tangent space of $M(\mathbf{x},0)$. The quantity $q(\mathbf{x})$ is useful for describing how a ball in $M(\mathbf{x},d)$ grows or shrinks with changes in t. If $q(\mathbf{x}) > 0$ then a ball on the manifold grows in volume and adjacent trajectories diverge. Likewise, if $q(\mathbf{x}) < 0$ a ball on the manifold shrinks and adjacent trajectories converge.

For certain values of $\mathbf{x}$ and ranges of d, $G(d,\mathbf{x})$ are local extrema of $q(\mathbf{x})$ along the manifold $M(\mathbf{x},d)$. This means that there are some system trajectories that are locally the maximally divergent or convergent trajectories in a region. The local minima of $q(\mathbf{x})$ are, qualitatively, representatives of a class of similar trajectories. Additionally, in two dimensions, the local maxima of $q(\mathbf{x})$ can act as boundaries between these classes.

State space reparameterization

State space can be reparameterized using coordinate systems embedded in the flow tangent manifolds. Given a manifold $M(x_0,d_0)$ any point in state space $\mathbf{x}_p$ can be parameterized using the coordinate vector $(\mathbf{s},d)$ where

$$ST(\mathbf{x}_p) = M(\mathbf{x}_0,d_0+d) + \int_0^{\mathbf{s}} v(ST^{-1}(\mathbf{s},d_0+d)\,d\mathbf{s} \qquad (3)$$

This parameterizes the point $\mathbf{x}_p$ in terms of the distance along the trajectory starting at $\mathbf{x}_0$ needed to reach the manifold that $\mathbf{x}_p$ lies on, and the distance along that manifold from $\mathbf{x}_p$ to the trajectory associated with $\mathbf{x}_0$. The space resulting from this reparameterization is called ’ST-space’. This reparameterization is shown visually in Fig.6.

III. Implementation

The design of the algorithm used to find the exemplar motions and state space reparameterizations defined in section II is not trivial. Observed policies contain noise and discretization artifacts that create unwanted extrema in the flow tangent divergence. Fig.2 illustrates the many local extrema that occur if the flow tangent divergence is computed directly from the observed policy. This section describes a method for robustly locating trajectories that lie on extrema of the flow tangent divergence.

Finding trajectories composed of local extrema of the flow tangent divergence, $q(\mathbf{x})$, is difficult because the numerical computation produces artifacts due largely to discretization of the policy. Numerical stability can be achieved by maximizing the curvature of quadratic approximations to the flow tangent manifolds integrated over a complete trajectory of the system. Although similar to the extrinsic curvature of $M(\mathbf{x},d)$, flow tangent divergence can be influenced by change in rate of flow along the manifold,

$$\frac{\partial |f(\mathbf{x},\mathbf{u}^*)|}{\partial v(\mathbf{x},\mathbf{u}^*)} \qquad (4)$$

while the curvature depends only on change in the direction of $f(\mathbf{x},u^*)$. This does not, however, affect the location of local extrema in the curvature. Therefore, both manifold curvature and flow tangent divergence can be used to find exemplar motions.

The curvature of the flow tangent manifold at a point $\mathbf{y}$ is approximated by fitting a second order polynomial to a ball in the flow tangent manifold containing $\mathbf{y}$, centered at $\mathbf{y}$, and of radius w. The radius w is chosen by searching for the largest value of w that does not cause the residual of the fit to exceed a threshold value. This procedure improves the quality of the metric as shown in Fig.3. The quadratic approximation can also be used to iteratively search for local extrema of the curvature along the manifold using a second order Newton-Raphson method. $\hat{q}(\mathbf{x})$ denotes curvature computed using this approximation.

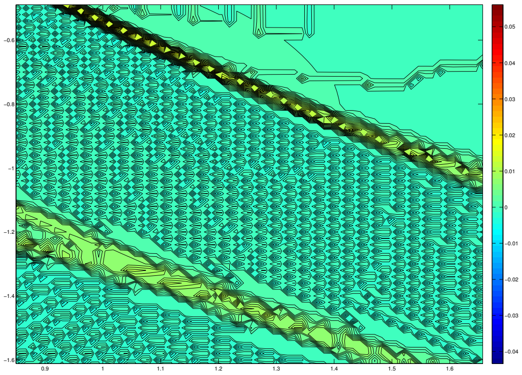

Fig. 2. Noise in the directly computed flow tangent divergence of system flow

Although quadratic approximation is effective for obtaining well behaved estimates of the curvature along the manifold, it is not sufficient for making stable estimates of exemplar motions. The exemplars are found by locating local extrema of the mean estimated curvature over trajectory segments on which the estimated curvature has uniform sign.

$$\underset{\mathbf{x}}{\operatorname{argmax}} \int_{t_0}^{t_f} \hat{q}(G(d,\mathbf{x}))\, ds \tag{5}$$

where t_0 and t_f are the minimum and maximum values for which

$$\begin{aligned} s_0 &\leq 0 \\ s_1 &\geq 0 \\ \hat{q}(G(d,\mathbf{x}))\hat{q}(G(0,\mathbf{x})) &> 0 \text{ when } s_0 \leq t \leq s_1 \end{aligned}$$

Exemplar trajectories are found by following the gradient of this metric along any flow tangent manifold.

IV. Example Problem

This section presents the application of our method to an example problem in two dimensions. The flow tangent manifolds, exemplar motions, and ST-space reparameterizations are shown. The system considered is a single link pendulum attempting to reach the unstable equilibrium pose $(\theta = 0 = 2\pi)$ with limited torque while minimizing distance to the goal state integrated over time. The center of the figure corresponds to the minimum energy state of the pendulum. The identification of task spines was previously considered in [13], although this work used a quadratic approximation to the value function for the computation.

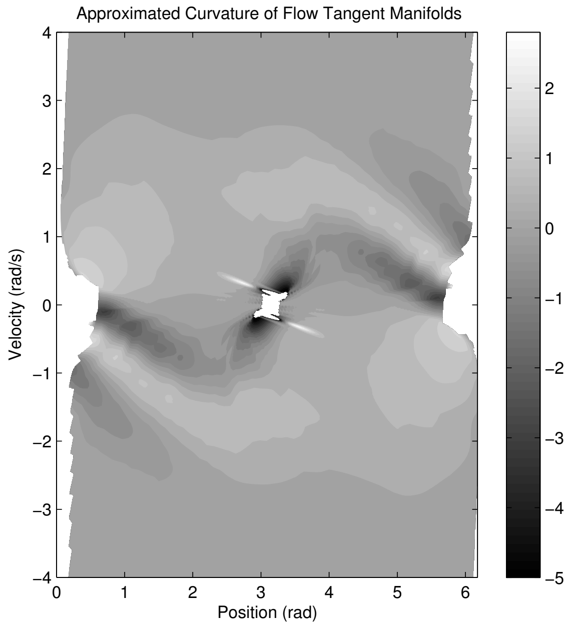

Fig. 3. Curvature computed from quadratic approximation of flow tangent manifolds

An optimal control policy was found via dynamic programming. The blue lines in Fig.4 show trajectories followed by the system using this controller. Visual inspection of the figure shows that there appear to be four qualitative trajectory classes in the system. These classes correspond to trajectories originating at the minimum energy state $(\pi, 0)$ and those originating at a high energy state $(*, \pm \text{inf})$. The trajectories originating from $(\pi, 0)$ can enter the goal from either the left or the right, resulting in a total of four expected trajectory classes terminating at the goal state.

The procedure described in section III is applied to this system, resulting in the flow tangent manifolds and exemplar motions shown in Fig.5. The exemplar motions identified by the system correspond to the four classes of motion identified by hand. One limitation visible in this result is that the system is not aware of unstable equilibria, in particular the two saddle points located approximately at $(0.7, 0)$ and $(2\pi - 0.7, 0)$.

Fig.6(a) shows a chosen trajectory, its associated flow tangent manifolds, and the intersections of neighboring trajectories with those manifolds. Fig.6(b) shows the mapping of this trajectory and its neighbors into the associated S-T space. In this space the manifold lines that are spaced at constant

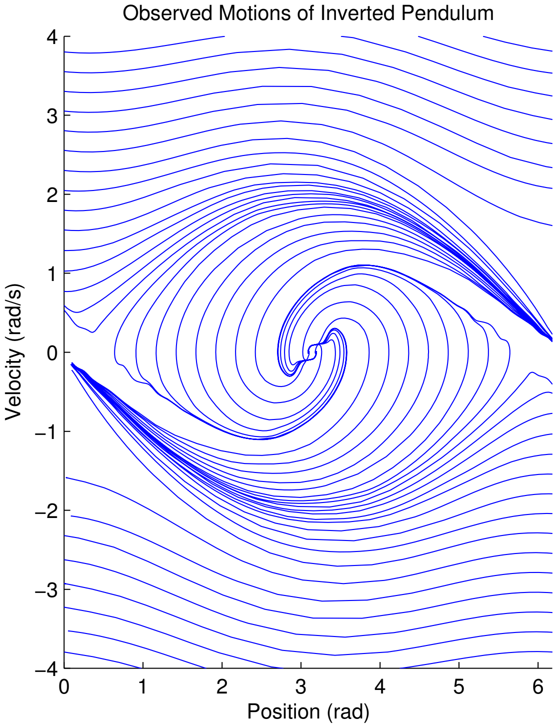

Fig. 4. Some observed trajectories of the controlled inverted pendulum.

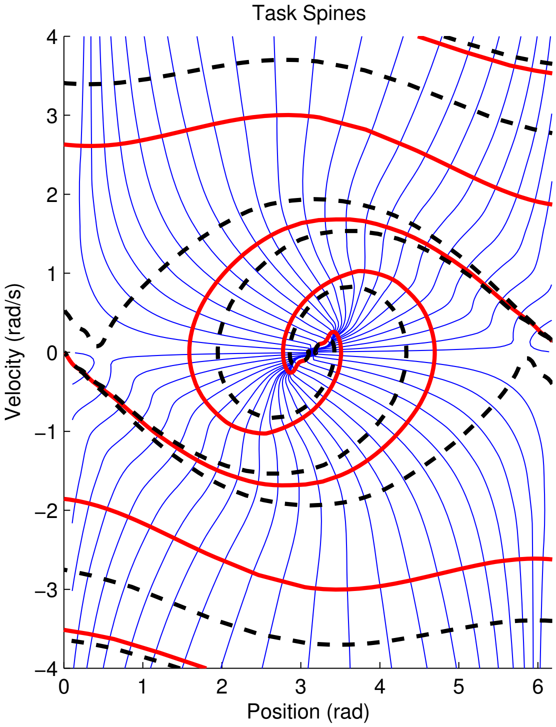

Fig. 5. Automatically identified spines of the inverted pendulum task. Solid lines are suggested boundaries, dashed lines are exemplar motions.

distance intervals along the key trajectory would appear as evenly spaced horizontal lines. The ST space provides an unambiguous mapping from points in state space to particular points on the key trajectory and, in this case, its behavior appears easily fit by a simple parametric model.

V. Discussion

We believe that control in ST-Space can allow linear controllers to operate over larger regions of state space. If the flow tangent manifolds about a trajectory $G(d,\mathbf{x})$ are well approximated by second order polynomials with coefficients linear in the distance along $G(d,\mathbf{x})$ then the direction of the ST-space flow will be linear in the coordinates:

$$f(\mathbf{x},u^*) \approx ST^{-1}(As) \tag{6}$$

The key limitation of our current approach lies in the method used for the identification of key trajectories. Our current method optimizes a scalar value — the curvature of the flow tangent manifold. While this works well for one-dimensional manifolds which arise in a two-dimensional state space, the notion of a single number for the curvature of a point in a higher-dimensional manifold is not well defined. One possible solution is to find trajectories that maximize the minimum curvature of the manifold in any direction.

In higher dimensions the choice of coordinates for the tangent spaces of flow tangent manifolds is undefined. Without a smooth function for determining the basis of the tangent space at each point on the manifold it is impossible to define the mapping from state space to ST space in general. If a coordinate system is defined for one manifold it can be propagated to other manifolds coherently by projecting a small ball on the original manifold to a new manifold using corresponding intersection points of observed trajectories. As the size of the ball on the original manifold decreases the transformation between the original and projected balls asymptotically approaches a linear transformation. This transformation can be applied to the basis of the tangent space at the center of the original ball to find the orientation of the basis of the tangent space at the center of the projected ball. This provides a useful definition of the basis of a tangent space on the new manifold.

Because the algorithm searches for the local maxima of a scalar function in a high dimensional space, it may be well suited to operation with higher dimensional systems. However, the cost of computing the approximated curvature of the flow tangent manifolds near a trajectory does grow exponentially in the number of dimensions.

VI. Future Work

We are currently exploring the identification of component strategies in human motion. It is widely believed that human motion, during balance and gait, for example, exhibits discrete

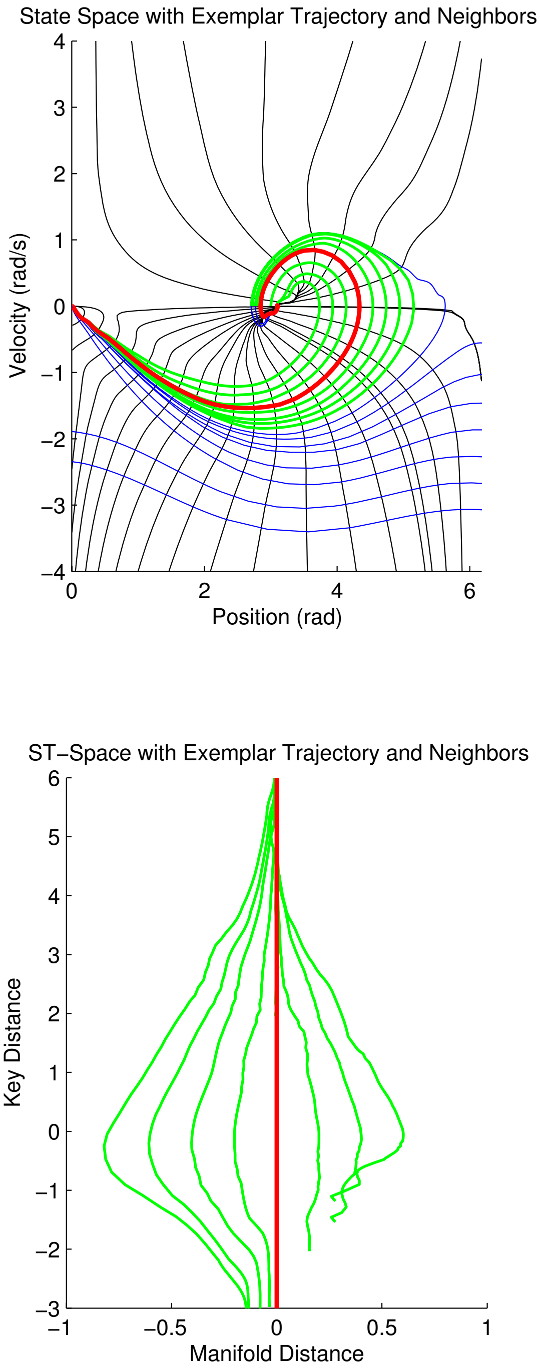

Fig. 6. top: An exemplar motion and its neighbors are shown intersecting flow tangent manifolds spaced uniformly along the exemplar. bottom: The ST-space mapping of the trajectories shown in the top illustration.

recovery strategies in response to unexpected perturbations [14]. For example, in response to a push, a person might raise their arms, or take a step forward, or move their hips, to stay in balance. We believe that our technique can help us automatically identify these discrete strategies, thereby helping us understand human balance better, as well as helping us control humanoid robots to balance better.

Working with human motion data requires a method that can operate in a state space with many dimensions. Furthermore, the number of available trajectory traces, say from human motion capture, populates this state space sparsely. These challenges require a reformulation of our approach to operate with a set of example trajectories where computing the derivative of the control on the manifold tangent to the trajectory is not possible, due to a sparse covering.

We are also interested in using this technique as part of an algorithm for actually finding optimal policies. The key bottleneck in the search for optimal policies is the curse of dimensionality — the search space grows exponentially in the number of state space dimensions. Hence, a good heuristic that identifies interesting (and conversely, boring) points in the state space where attention, and hence computation, can be focused (or defocused) is crucial. Because the behavior of trajectories in a region around the exemplar trajectories is characterized by convergence to that trajectory, we believe that linear controllers will perform well in these regions, making the points near an exemplar trajectory not very interesting. The boundaries between trajectory classes could potentially be used in a fashion analogous to the simplex method introduced by [9].

VII. ACKNOWLEDGMENTS

The authors gratefully acknowledge the contributions of Chris Atkeson, Geoffrey Gordon, Jessica Hodgins, and Arash Mahboobin. The material is based upon work supported in part by the National Science Foundation under NSF Grants ECS-0325383, CNS-0224419, and DGE-0333420. Stuart Anderson was partially supported by Intel Research Pittsburgh.

REFERENCES

[1] M. Mason, "The mechanics of manipulation," in *Robotics and Automation. Proceedings. 1985 IEEE International Conference on,*, vol. 2, 1985, pp. 544–548.

[2] M. Mataric, M. Williamson, J. Demiris, and A. Mohan, "Behavior-based primitives for articulated control," in *Fifth Interational Conference on Simulation of Adaptive Behavior, Proceedings of.* MIT Press, 1998, pp. 165–170.

[3] A. Fod, M. Mataric, and O. Jenkins, "Automated derivation of primitives for movement classification," *Autonomous Robots*, vol. 1, pp. 39–54, January 2002.

[4] O. Jenkins and M. Mataric, "Automated derivation of behavior vocabularies for autonomous humanoid motion," in *Autonomous Agents and Multi Agent Systems, Proceedings of.*, 2003, pp. 225–232.

[5] M. Branicky, T. Johansen, I. Petersen, and E. Frazzoli, "On-line techniques for behavioral programming," in *Proceedings of the 39th IEEE Conference on Decision and Control*, 2000, pp. 1840–1845.

[6] D. Bentivegna and C. Atkeson, "Learning from observation using primitives," in *IEEE International Conference on Robotics and Automation, Proceedings of.*, 2001, pp. 1988–1993.

[7] M. Erdmann and M. Mason, "An exploration of sensorless manipulation," *IEEE Journal of Robotics and Automation*, vol. 4, no. 4, August 1988.

[8] R. Abelson, M. Eisenberg, M. Halfant, J. Katzenelson, E. Sacks, G. Sussman, J. Wisdom, and K. Yip, "Intelligence in scientific computing," *Communications of the ACM*, vol. 32, no. 5, pp. 546–562, 1989.

[9] F. Zhao, "Extracting and representing qualitative behaviors of complex systems in phase space," *Artificial Intelligence*, vol. 69, pp. 51–92, 1994.

[10] ——, "Automatic analysis and synthesis of controllers for dynamical systems based on phase-space knowledge," MIT Artificial Intelligence Laboratory, Technical Report 1385, 1992.

[11] N. Roy, G. Gordon, and S. Thrun, "Finding approximate pomdp solutions through belief compression," *Journal of Artificial Intelligence Research*, vol. 23, pp. 1–40, 2005.

[12] J. Bagnell, S. Kakade, A. Ng, and J. Schneider, "Policy search by dynamic programming," *Proc. NIPS*, 2001.

[13] C. G. Atkeson, "Using local trajectory optimizers to speed up global optimization in dynamic programming," in *Advances in Neural Information Processing Systems*, J. D. Cowan, G. Tesauro, and J. Alspector, Eds., vol. 6. Morgan Kaufmann Publishers, Inc., 1994, pp. 663–670. [Online]. Available: citeseer.ist.psu.edu/atkeson94using.html

[14] R. Cham and M. Redfern, "Lower extremety corrective reactions to slip events," *Journal of Biomechanics*, vol. 34, no. 11, pp. 1439–1445, November 2001.