Manipulation Gaits: Sequences of Grasp Control Tasks

Robert Platt Jr. Andrew H. Fagg Roderic A. Grupen

Laboratory for Perceptual Robotics
Department of Computer Science
University of Massachusetts, Amherst
{rplatt, fagg, grupen}@cs.umass.edu

Abstract

In dexterous manipulation, an object must be reconfigured while maintaining a stable grasp. This may require that the object be re-grasped in order to avoid finger workspace limits. In this paper, we present a set of closed-loop controllers designed to achieve force-related objectives such as wrench closure, and show how they may be concurrently combined. Furthermore, we show that dexterous manipulation behavior may be generated by sequencing concurrent combinations of these controllers. We show that dexterous manipulation can be viewed as a task that is accomplished in the context of a wrench closure constraint. We hypothesize this approach can generalize to any task that must be accomplished while maintaining a set of constraints.

1 Introduction

Many common robotics problems have a multi-objective nature. For example, bipedal walking requires that the robot translate while maintaining its balance. Teams of mobile robots may be required to explore while maintaining specific formation or line-of-sight constraints. In manipulation, the robot is required to achieve a particular hand/object configuration while maintaining a grasp.

In each of these problems, the desired behavior must be performed while maintaining a set of task-specific constraints. In this paper, we describe a control framework that enables a robot to explore different behaviors without violating a specified constraint. We explore this in the context of dexterous manipulation.

Several approaches have been proposed for solving manipulation tasks using geometric primitives. In these approaches, each primitive executes a fixed, coordinated trajectory in configuration space that is based on geometric assumptions about the object and manipulator. For example, Michelman and Allen used several primitives, including a "log rolling" strategy to unscrew a child-proof bottle cap [7]. Han and Trinkle identify two finger gaiting strategies designed for a three-fingered manipulator that require prior object knowledge [3].

Huber generalized this perspective to incorporate closed-loop controllers in a "control basis" rather than using geometric primitives [5]. Control synthesis is viewed in terms of combinations of control objectives rather than combinations of geometric artifacts. Huber showed that finger gaiting behavior could be created by sequencing such controllers and demonstrated that policies derived this way generalize to different robots including a quadruped walking platform.

Closely related to the approach advocated in this paper is Burridge, Rizzi, and Koditschek's research into controller sequencing and funneling [1]. In that work, closed-loop controllers were sequenced in order to make the system robust in a broad region of state space. In our current work, we combine and sequence controllers so as to maintain control within a set of constraints.

This paper describes a control basis that is appropriate for force-based interactions using whole body contacts. We show how the force-based controllers in this control basis may be concurrently combined to aid in robustly navigating through a space of different wrench closure configurations. This work is based on grasp controllers that have equilibria in wrench closure configurations. We show that these controllers can be used to generate manipulation policies that maintain a wrench closure constraint. This approach is demonstrated on Dexter, the UMass humanoid robot.

Section 2 gives an overview of the control basis framework. Section 3 describes the controllers used in this work, and section 4 presents methods for concurrently combining force-based controllers. Section 5 describes how manipulation problems can be solved using a markov decision process, and section 6 presents two experiments that validate the approach on Dexter.

2 The Control Basis Approach

The control basis approach is a framework for combining closed loop controllers in a systematic way to accomplish a variety of different behavioral objectives. In this framework, a wide variety of behavior is described by controllers derived from a small set of potential functions. Controllers belonging to a control basis typically implement fundamental behaviors related to force, motion, and kinematic objectives.

Controllers are generated by parameterizing (binding) a potential function Φ_i with a set of input resources (sensors) σ and output resources (effectors) τ. This binding is denoted: $\Phi_i|_\tau^\sigma$. The parameterized controller uses control resources τ to descend the potential function parameterized by input resource σ. For example, $\Phi_m|_l^{\vec{ref}}$ descends a motion control potential function parameterized by a target reference location. This controller moves the left hand toward the location specified by $\vec{ref}$.

In this framework, we allow two or more controllers to execute concurrently in a prioritized manner. We use the term "subject to" ("$\triangleleft$") to describe this combination. We say that Φ_S executes "subject to" Φ_P ($\Phi_S \triangleleft \Phi_P$) when Φ_S is constrained to operate exclusively in the nullspace of Φ_P. For example, we would write $\Phi_k \triangleleft \Phi_m$ to describe a composite controller that optimizes kinematic configuration in the nullspace of a motion objective. Similarly, a motion objective can be specified to operate in the nullspace of wrench closure: $\Phi_m \triangleleft \Phi_g$. This control combination moves the arm toward a reference position while maintaining wrench closure on an object.

Closed-loop controllers and combinations of controllers can be sequenced to generate a variety of robot behavior. Huber showed that behavior can be explored in the context of a Markov Decision Process (MDP) [4]. An MDP is a framework for modeling stochastic control problems. There are states and actions. Each state is paired with a set of allowable actions. When an action from this set executes, the state of the system changes according to a fixed, but possibly unknown, probability distribution. In the control basis approach, an MDP models the evolution of a set of artificial potentials in the context of controller execution. For our purposes, the state of a potential function can often be succinctly captured by a bit that indicates whether that function is converged or not. The state of the system is represented as a vector of these convergence indicator bits. One of the advantages of using an MDP is that a number of machine learning techniques exist for finding policies (action sequences) on-line. Reinforcement learning (RL) is a powerful on-line control algorithm that learns through trial-and-error [11]. Huber showed that the control basis framework can accelerate learning times required for RL to converge.

3 Artificial Potential Functions

In the control framework, all controllers are drawn from a set of artificial potential functions known as a control basis. In this paper, we focus on a control basis appropriate for describing force-based behaviors such as grasping and manipulation.

The grasp control artificial potential Φ_g descends

$$\epsilon_w = \vec{w}^T \vec{w}, \quad \vec{w} = \sum_{1 \le i \le n} \vec{w}_i \tag{1}$$

where $\vec{w}_i$ is the wrench applied at the i^{th} contact [2, 8]. This control law converges when the net wrench applied by the contacts is zero. In the presence of friction, such a grasp achieves wrench closure because it fulfills the conditions for non-marginal equilibrium. Non-marginal equilibrium requires the contact forces achieving net zero force lie strictly inside their corresponding friction cones and has been shown to be a sufficient condition for wrench closure [10]. This is equivalent to the grip Jacobian having a nullspace.

In order to descend ϵ_w, the controller decomposes the wrench error into force error and moment error components $\epsilon_f = \vec{f}^T \vec{f}$ and $\epsilon_m = \vec{m}^T \vec{m}$. The two error gradients are calculated separately: $\frac{\partial \epsilon_f}{\partial \vec{x}} = \frac{\partial \epsilon_f}{\partial \vec{f}} \frac{\partial \vec{f}}{\partial \vec{x}}$ and $\frac{\partial \epsilon_m}{\partial \vec{x}} = \frac{\partial \epsilon_m}{\partial \vec{m}} \frac{\partial \vec{m}}{\partial \vec{x}}$. The controller descends the wrench error gradient by projecting the moment error gradient into the nullspace of the force error gradient as follows:

$$\frac{\partial \epsilon}{\partial \vec{x}} = \frac{\partial \epsilon_f}{\partial \vec{x}} + \mathcal{N}\left(\frac{\partial \epsilon_f}{\partial \vec{x}}\right) \frac{\partial \epsilon_m}{\partial \vec{x}} \tag{2}$$

This formulation reduces the number of spurious local minima that can trap the controller. For more information on this approach, see [8].

Effective grasp control requires a constant stream of information regarding the surface normal perceived at each contact. The contact control artificial potential, Φ_c, ensures that this information is available by placing the appropriate contacts on the object. Force control is used to exert a small force along the inward surface normal of each contact. If a contact momentarily drifts away from the surface, this force controller restores contact. In the implementation reported in this paper, force control is restricted to the finger flexion degrees of freedom. The arm degrees of freedom are controlled separately to condition the finger so that when it does make object contact, the contact point is well within the fingertip's workspace.

The motion control artificial potential Φ_m descends $\epsilon_m = (\vec{x}_{goal} - \vec{x}_{current})^2$ where $\vec{x}_{goal}$ and $\vec{x}_{current}$ are the goal and current manipulator positions respectively. The kinematic artificial potential, Φ_k, used in the current work descends $\epsilon_k = (\vec{q}_{center} - \vec{q}_{current})^2$. In this equation, $\vec{q}_{center}$

is the vector of joint angles at the center of their range, and $\vec{q}_{current}$ is the current configuration of joint angles.

3.1 Binding the Grasp Controller to Whole Body Contacts

Φ_g can be instantiated with a non-unitary subset of the input resources σ (contacts used to compute the artificial potential) and a subset of the output resources τ (contacts that move in response to the control output). For example, suppose $\{1, 2, 3\}$ is the set of fingertip contact resources on a three-fingered hand. Then, $(\Phi_g|_{1,2}^{1,2,3})$ displaces contacts 1 and 2 in order to form wrench closure among contacts 1, 2, and 3.

Grasp controllers can also be parameterized by *virtual contacts* (virtual fingers). A virtual contact is a set of contacts that provide a single oppositional wrench equal to the net wrench applied by the constituent contacts [6, 9]. External forces on the object, such as gravity, are also considered virtual contacts. Gravity can be considered a contact whose position is the center of mass of the object and whose magnitude is equal and opposite to the net contact force. We refer to a grasp utilizing different contact types a *whole body grasp.*

For example, Dexter, the UMass Humanoid, consists of two arms, two hands, and a stereo vision head as shown in Figure 1. If the set of fingertip contacts on each hand are combined to form a single virtual contact, the pair of arms can be considered to be a pair of "fingers." One possible set of contact resources on this platform is $\{l, r, g\}$ where $\{l\}$ is the virtual contact on the left hand, $\{r\}$ is the virtual contact on the right hand, and $\{g\}$ is the virtual contact corresponding to gravity. For example, Figure 1 shows Dexter holding a ball in wrench closure between virtual contacts on the left and right hands using controller $(\Phi_g|_{l,r}^{l,r})$. Figure 2 shows Dexter holding a ball using both $(\Phi_g|_{l,r}^{l,r})$ and $(\Phi_g|_{l}^{l,g})$.

4 Using the Nullspace to Maintain Constraints

Many robotics tasks require a desired objective to be accomplished in the context of a set of constraints. In the manipulation domain, such a problem involves realizing a specified object reconfiguration over time while maintaining wrench closure. In this section, we show how grasp controllers may be concurrently combined to maintain a wrench closure constraint. We also formulate an MDP that enables the system to evaluate the relative value of control decisions that traverse the set of wrench closure states.

4.1 Maintaining Wrench Closure Constraints

When a secondary control action (Φ_S) is executed in the nullspace of an existing wrench closure condition, the secondary control action is guaranteed not to disturb existing wrench closure. Implementing this relationship is trivial if the output resources of Φ_S are independent of the resources used to maintain wrench closure. For example, this is true when executing $\Phi_g|_{r}^{l,r} \triangleleft \Phi_g|_{l}^{l,g}$. In this case, it is safe to execute both controllers concurrently because they operate on different sets of effectors.

However, when Φ_S requires resources that are also used to maintain wrench closure, it is necessary to project the control action specified by Φ_S into the nullspace of the wrench closure objective. This happens when the wrench closure condition exists between the left and right hands. Assuming that the primary wrench closure objective has already been achieved by a prior run of a grasp controller, maintaining wrench closure only requires force control. Each contact must exert the appropriate grasping force normal to the object surface. The secondary objective can be projected into the nullspace of this force control objective.

In the current work, we simplify the force control problem by assuming that the force control objective requires the contacts to make only small motions relative to each other. If this is the case, force control can be accomplished exclusively by the finger flexion degrees of freedom while the hands maintain a constant relative pose. This assumption makes it easier to project the secondary objective into the nullspace of a control objective that maintains the constant relative pose.

For example, suppose that the robot is already holding the ball between left and right hands as shown in Figure 1 and we now execute $\Phi_m|_{l}^{\vec{ref}}$ subject to $\Phi_g|_{l,r}^{l,r}$. Let $\frac{\partial \epsilon_m}{\partial \vec{q}}$ be the gradient associated with $\Phi_m|_{l}^{\vec{ref}}$, and let $\frac{\partial \epsilon_r}{\partial \vec{q}}$ be the gradient that maintains a constant relative pose between the two hands. These two objectives can be combined by projecting $\frac{\partial \epsilon_m}{\partial \vec{q}}$ into the nullspace of $\frac{\partial \epsilon_r}{\partial \vec{q}}$ as follows:

$$\frac{\partial \epsilon}{\partial \vec{q}} = \frac{\partial \epsilon_r}{\partial \vec{q}} + \mathcal{N}\left(\frac{\partial \epsilon_r}{\partial \vec{q}}\right)\frac{\partial \epsilon_m}{\partial \vec{q}} \quad (3)$$

This results in the left hand moving toward $\vec{ref}$ while the right hand follows so as to maintain wrench closure. If it is not possible for the left hand to reach $\vec{ref}$, the system will reach as far as it can and then stop. See Figure 1. A similar approach can be used if Φ_S is a grasp controller that uses gravity as a resource such as $\Phi_g|_{r}^{r,g}$.

4.2 Maintaining Contact Constraints

The grasp controller requires that the object remains within reach. However, the grasp controller displaces contacts along the object surface without guaranteeing that the object surface remains within reach. Therefore, a

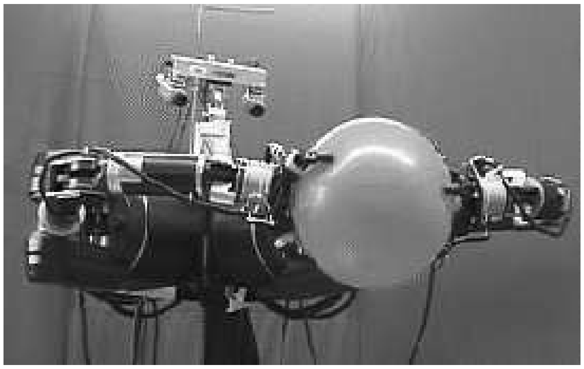

Figure 1: Dexter reaching to the left while grasping ball with $\Phi_g|_{l,r}^{l,r}$.

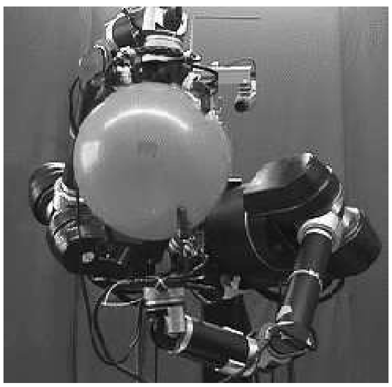

Figure 2: Dexter grasping a ball with both $\Phi_g|_{l,r}^{l,r}$ and $\Phi_g|_{l}^{l,grav}$.

mechanism is needed to ensure that the hand does not drift away from the object surface.

The grasp controller can be used together with the contact controller to solve this problem. The contact controller ensures that each contact remains on the surface of the object. Since the grasp controller requires good contacts, the contact controller funnels the state of the system toward configurations where the grasp controller can function. These controllers can be concurrently combined by projecting the grasp control gradient ($\frac{\partial \epsilon_w}{\partial \vec{q}}$) into the nullspace of the contact control gradient ($\frac{\partial \epsilon_c}{\partial \vec{q}}$) using an equation similar to Equation 3. In the notation of the control basis, this is represented by: $\Phi_g \triangleleft \Phi_c$. The controller that results from this control composition slides the contacts along the surface of the object in the direction of the negative wrench error gradient. As the contacts move, the arm moves to keep the fingers in contact.

5 Maintaining Constraints in an MDP

In the course of manipulation tasks, it is often necessary to maintain wrench closure on an object over the course of many actions.

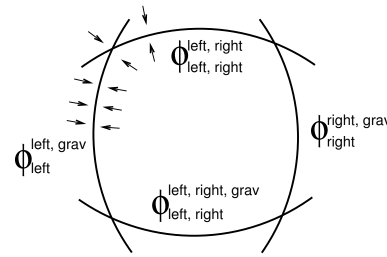

Figure 3: Abstract representation of the equivalence class of wrench closure states. Each of the four arcs represents a wrench closure manifold in the robot's multi-dimensional configuration space. A manipulation policy moves the state of the system along the arcs to the desired grasp.

Figure 3 shows a two-dimensional abstract representation of configuration space. The four arcs represent lower dimensional manifolds in the configuration space due to four grasp controllers. At any point in time, the state of the robot is a point in the diagram. When a controller is activated, the point-robot is drawn to the equilibrium manifold corresponding to a particular wrench closure solution. If a subordinate controller is executed in the nullspace of the first controller, the state of the system moves along the first controller's equilibrium manifold toward configurations that also satisfy the subordinate controller. Under the right conditions, this can yield two simultaneous wrench closure solutions.

We use an MDP like the one shown in Figure 4 to represent the transition dynamics over grasp states and actions. In the Figure, each state is drawn as a circle that corresponds to the condition that a stable grasp exists among the pairs of contacts listed in the circle. The arrows that fan out from a particular state represent the set of available control actions that move the system through the space of grasp states. For every state, actions that do not maintain at least one wrench closure condition asserted in that state are pruned from the MDP. This simple rule guarantees that the MDP only represents manipulation policies that maintain wrench closure.

Notice that state in the MDP is not a geometrical assertion, but a report about the membership of the grasp in one or more of the manifolds illustrated in Figure 3. Control actions in the MDP transition the system from one type of grasp to another by using subordinate controllers to slide along wrench closure manifolds. This approach can preserve the wrench closure guarantee while continuously adjusting the grasp to accommodate a task.

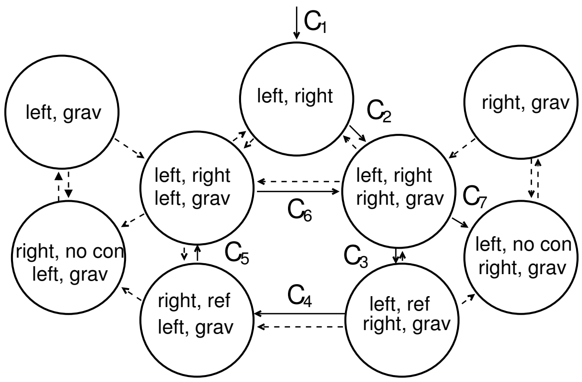

Figure 4: Illustration of the grasp MDP defined over a space of wrench closure conditions. Each circle represents the state when wrench closure exists among the resources listed in the circle. The arrows pointing between the circles represent possible actions that move the system between one state and another.

6 Experiments

The control basis approach has been explored on a number of platforms including a quadruped walking robot [4], multiple coordinated hand/arm systems [2], distributed mobile robots, and Dexter, the UMass humanoid. The current work is demonstrated on Dexter. This platform consists of two Barrett WAMs (Barrett Technologies, Cambridge MA) mounted on a humanoid frame. A BiSight (stereo vision) "head" is mounted on top of the frame to provide visual feedback. Each WAM is equipped with a 3-finger, 4 DOF Barrett Hand. Mounted on the tip of each Barrett hand finger is an ATI 6-axis force-torque sensor that makes possible the computation of a fingertip contact location and normal. We demonstrate the expressiveness and flexibility of the force-based control basis described in this paper in two experiments where Dexter learned two different manipulation policies using the same set of control primitives.

In the first experiment, a policy for translating a large ball (18 centimeters in radius) approximately 80 centimeters from the medial plane was learned autonomously using SARSA(λ) reinforcement learning (an online form of MDP dynamic programming) [11]. The task starts with Dexter holding the ball between two hands. The robot is required to translate the ball. However, it is not possible to move the ball directly to the goal position because the arms cannot reach far enough to maintain the two-handed grasp during the motion. Therefore, the robot must first execute a regrasp and then move the ball. An MDP similar to the one in Figure 4 models the transition dynamics of the controllers. The robot receives reward only after achieving the required ball displacement. The regrasp necessary to solve the task was learned autonomously through trial-and-error exploration. The structure of the MDP ensured that only

Step	Label	Composite Controller
1	C_1	$\Phi_g\|_{l,r}^{l,r} \triangleleft \Phi_c\|_{l,r}^{l,r}$
2	C_2	$\Phi_g\|_{r}^{r,g} \triangleleft \Phi_g\|_{l,r}^{l,r}$
3	C_7	$\Phi_c\|_{l}^{l} \triangleleft \Phi_g\|_{r}^{r,g}$
4		$\Phi_m\|_{r}^{\vec{ref}} \triangleleft \Phi_g\|_{r}^{r,g}$

Table 1: A sequence of actions that translates the large ball. The label C_i references each action to an arrow in Figure 4. In step 1, the robot grasps the ball with two hands. In steps 2 and 3, the necessary remanipulation is executed and the ball is placed in the right hand. Finally in step 4, the robot executes the desired reaching command.

control actions that maintained wrench closure were considered. This made learning safe and efficient. An optimal trajectory through the learned policy is shown in Table 1.

In the second experiment, a simulation of Dexter learned to rotate a large ball 180 degrees. The resulting policy was executed on the robot. As before, SARSA(λ) reinforcement learning was used. However, in this experiment, the learning system supplemented its on-line experience with model-based knowledge also acquired on-line. This task starts with the robot holding the ball between its arms. The system is rewarded for achieving a ball orientation 180 degrees away from the starting orientation. Because of workspace limitations, it is not possible for the re-orientation to occur without a regrasp. The system learns a valid regrasp sequence that achieves the re-orientation. The MDP shown in Figure 4 constrains the system's exploration to only those control actions that maintain wrench closure.

The learning process ran for a total of 40 episodes. On each episode, the learning system executed actions until the goal was reached or for a maximum of 15 steps. Figure 5 shows a learning curve that plots the number of steps required to reach the goal as a function of episode number. As the number of episodes increases and learning continues, the average policy learned moves closer and closer to optimal.

These two experiments demonstrate that the control basis approach is able to represent these two different behaviors using a single control substrate. The ability to represent a variety of behavior in a single framework is a critical prerequisite for autonomously learning task-appropriate behavior. As the number of controller parameterizations and legal controller compositions grows, we expect the variety of represented behaviors to rise dramatically.

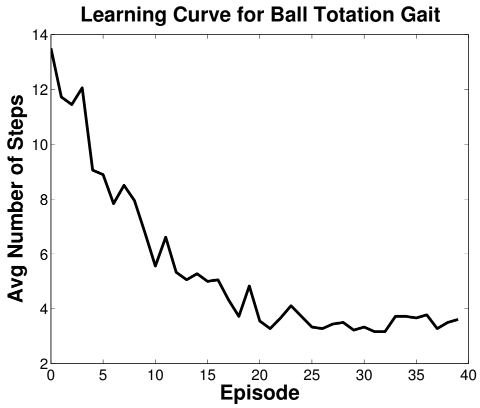

Figure 5: Learning curve for learning the rotation gait. Data is averaged over 18 separate learning experiments. The horizontal axis is episode number in the learning process. The vertical axis is the average number of control actions required to complete the task in that episode. As learning progresses, the number of actions required to rotate the ball drops quickly.

7 Conclusion

In this paper, we describe a control basis capable of generating a variety of force-based interaction. Among the controllers in this basis, this paper focuses on the grasp and contact artificial potentials. We show that these two force-based artificial potentials can be parameterized with arbitrary sets of whole body contact resources and combined in a variety of interesting ways. The abundance of control behavior expressed by different parameterizations and combinations of these artificial potentials results in a flexible system for generating task-level behavior.

We show that this control basis is well-suited to force-based constraint maintenance problems. This approach is demonstrated in the context of dexterous manipulation. A markov decision process (MDP) is defined over the space of wrench closure conditions. A simple rule prunes all actions that do not maintain wrench closure. The resulting MDP can be used by reinforcement learning techniques to autonomously learn manipulation policies that accomplish specified goals. We present two demonstrations of this technique where manipulation gaits were autonomously learned.

We expect this approach to generating dexterous manipulation in the context of a wrench closure constraint to generalize to arbitrary tasks where a set of constraints must be maintained. Aspects of this approach have already been demonstrated in a mobile robot exploration task where robots are constrained to maintain line-of-sight [12].

In future work, we hope to use this force-based control basis to describe block building tasks. We hypothesize that the same wrench closure principles that underlie dexterous manipulation are also adequate to describe the stability of block structures.

Acknowledgments
The authors which to thank the members of the Dexterous Robotics Laboratory at Johnson Space Center for their support. This work was supported by DARPA, NASA grant NAG9-1445, NASA Graduate Student Researchers Program Fellowship NGT 9-61 2, and NSF grant EIA-9703217.

References

[1] R. Burridge, A. Rizzi, and D. Koditschek. Sequential composition of dynamically dexterous robot behaviors. *Int. J. Rob. Res.*, 18(6), 1999.

[2] J. Coelho and R. Grupen. A control basis for learning multifingered grasps. *Journal of Robotic Systems*, 1997.

[3] L. Han and J. Trinkle. Dextrous manipulation by rolling and finger gaiting. In *IEEE Int'l Conf. Robotics Automation*, 1998.

[4] M. Huber. *A Hybrid Architecture for Adaptive Robot Control.* PhD thesis, U. Massachusetts, 2000.

[5] M. Huber and R. Grupen. Robust finger gaits from closed-loop controllers. In *IEEE Int'l Conf. Robotics Automation*, 2002.

[6] C. MacKenzie and T. Iberall. *The Grasping Hand.* North-Holland, 1994.

[7] P. Michelman and P. Allen. Forming complex dextrous manipulations from task primitives. In *IEEE Int'l Conf. Robotics Automation*, 1994.

[8] R. Platt, A. H. Fagg, and R. A. Grupen. Nullspace composition of control laws for grasping. In *IEEE Int'l Conf. on Intelligent Robots and Systems*, 2002.

[9] R. Platt, A. H. Fagg, and R. A. Grupen. Extending fingertip grasping to whole body grasping. In *IEEE Int'l Conf. Robotics Automation*, 2003.

[10] J. Ponce, S. Sullivan, A. Sudsang, J. Boissonnat, and J. Merlet. On computing four-finger equilibrium and force-closure grasps of polyhedral objects. *Int. J. Rob. Res.*, 1996.

[11] R. Sutton and A. Barto. *Reinforcement Learning, An Introduction.* MIT Press, 1998.

[12] J. Sweeney, TJ Brunette, Y. Yang, and R. A. Grupen. Coordinated teams of reactive mobile platforms. In *IEEE Int'l Conf. Robotics Automation*, 2002.