Control Pre-Imaging for Multifingered Grasp Synthesis*†

J. A. Coelho Jr. R. A. Grupen

Laboratory for Perceptual Robotics
Department of Computer Science
University of Massachusetts, Amherst, 01003

Abstract

This paper discusses the issues involved in developing a grasp controller, within the framework of control composition, and introduces control pre-imaging. Pre-imaging is a design technique for augmenting the performance of a baseline controller through the opportune activation of metalevel control actions. Such metalevel control actions are activated whenever the current system state is associated to past unsuccessful patterns of interaction between the existing predictable, stable controller and the environment. The technique is illustrated by the implementation of a grasp controller capable of generating grasp configurations through successive, local refinements, given the position and normal of each contact.

1 Introduction

The grasp synthesis or grasp planning problem has been studied extensively. Most published work concerns off-line grasp synthesis, and two general approaches have been used, namely geometric force closure grasp synthesis [4, 5, 11], and optimization-based grasp synthesis [1, 7, 8]. Both approaches aim at finding the best grasp configuration under a certain grasp metric, either (1) purely based on object geometry or (2) based on specific aspects of grasp configurations. Some approaches restrict the number of contacts according to the object dimensionality (2 contacts in 2D and 3D [1], 2 contacts in 2D [4], at least 3 contacts in 3D and 2 contacts in 2D [8]), others constrain the object representation (polygons or polyhedra are assumed in [5, 11], and smooth/parametric models are required in [1, 4, 7]).

None of the approaches above yields a grasp controller. Grasps are preceded by off-line computation of the best grasp configuration, and the algorithmic complexity can be very high [4] or the computation may involve exhaustive search over the object geometry [5, 11]. There is little concern about *how* to generate the final contact configuration; presumably, finger positioning can be executed by an open-loop control policy or by special purpose manipulators. With the exception of [8], no other approach allows for incremental models of the object. A complete model (either in terms of its real geometry or in terms of a smooth model) is required before grasp synthesis starts, and it is not clear how incremental object models could be accommodated in the framework of these approaches.

This paper addresses the development of robust controllers for multifingered grasps of convex objects. We review the force closure (FC) and the moment closure (MC) controllers [2, 6] (see also [3]) used collectively to effect wrench closure contact geometries using frictionless, kinematically unconstrained point contacts. We then focus the body of this paper on supervising their composition to avoid failures in terms of unsatisfactory grasp configurations.

2 Grasp control composition

Our primary objective is to compose two elemental controllers (the FC and MC controllers) into an efficient grasp controller. Both controllers are designed to maximize grasp stability, based on local estimates of the wrench closure error surface.

The null space of the grasp matrix defines a basis for the perturbation wrenches that can be resisted by a given contact configuration. Both controllers target the construction of a null space within the grasp matrix through minimization of the residual wrench $\vec{\rho}$

$$\begin{aligned}\epsilon &= \vec{\rho}^T \vec{\rho} \\ &= \left(\vec{t} - \frac{1}{n}\sum_{i=1}^{n} \hat{\omega}_i\right)^T \left(\vec{t} - \frac{1}{n}\sum_{i=1}^{n} \hat{\omega}_i\right), \qquad (1)\end{aligned}$$

*This work is supported in part by NSF CDA-8922572, IRI-9116297, IRI-9208920, and CNPq 202107/90.6.

where $\vec{\rho}$ expresses the net wrench over $1 \leq j \leq n$ contacts, $\vec{t}$ is a user-specified wrench closure *bias* (for the null grasp task, $\vec{t} = \vec{0}$), and $\vec{\omega}_i$ is the wrench vector resulting from the i^{th} interaction force. The elements $t_j \in [-1, 1]$ of $\vec{t}$ and the elements $w_{ij} \in [-1, 1]$ of $\vec{\omega}_i$ are qualitative in the sense that they do not reflect engineering units of force and torque, but express the relative ability of a contact configuration to transmit forces and torques through the object's surface.

Each controller assumes a distinct wrench domain model of the object geometry, or how contact forces are locally transformed into wrenches on the object frame. No explicit object model is ever required – control actions are computed from local sensory information, namely position and normal of each contact.

2.1 Force closure controller

The FC controller [6] is used to navigate frictionless point contacts on the continuous Gaussian sphere. This simplified system uses only relative contact position to eliminate the force closure component of the residual (Equation 1). If θ and ϕ are the angular coordinates on the Gaussian sphere, then the corresponding wrench domain model of the object $\vec{W}(\theta, \phi) = [F_x\ F_y\ F_z\ M_x\ M_y\ M_z]$ becomes

$$\begin{array}{rclcrcl} w_1 &=& f_x = -cos(\theta)cos(\phi) & \quad & w_4 &=& m_x = 0 \\ w_2 &=& f_y = -sin(\theta)cos(\phi) & & w_5 &=& m_y = 0 \\ w_3 &=& f_z = -sin(\phi) & & w_6 &=& m_z = 0 \end{array}$$

The error metric ϵ_{FC} is obtained by substituting the above wrench model on Equation 1. ϵ_{FC} can be demonstrated to be unimodal; therefore, a globally convex controller based on the gradient $\frac{\partial \epsilon_{FC}}{\partial \theta_i}$ can be derived so as to minimize the original error metric ϵ_{FC}.

2.2 Moment closure controller

The moment closure (MC) controller is based on the assumption that wrenches are produced as if the contact were in a planar region on the object surface – the contact plane. The forces transmitted through the contact plane are constant at all positions on that plane. The moments applied to the object, $\vec{m} = \vec{r} \times \vec{f}$, vary linearly with surface coordinate and pass through zero where the perpendicular passes through the plane. The resulting wrench model is:

$$\begin{array}{rcl} w_1 &=& f_x = -cos(\theta_0)cos(\phi_0) \\ w_2 &=& f_y = -sin(\theta_0)cos(\phi_0) \\ w_3 &=& f_z = -sin(\phi_0) \\ w_4 &=& m_x = -r_\theta sin(\phi_0)cos(\theta_0) + r_\phi sin(\theta_0) \\ w_5 &=& m_y = -r_\theta sin(\phi_0)sin(\theta_0) - r_\phi cos(\theta_0) \\ w_6 &=& m_z = r_\theta cos(\phi_0) \end{array} \quad (2)$$

where (r_θ, r_ϕ) are the surface coordinates of the contact in the plane. The error ϵ_{MC} is multimodal, having many local minima; the MC controller is based on a minimum-preserving convex approximation of ϵ_{MC}.

2.3 Control composition

The FC and MC controllers can be combined into a grasp controller proved to be complete for the case of two contact, null grasp of regular polygons [2]. These controllers constitute a *basis* for the construction of grasp controllers. If α_{FC} and α_{MC} are non-negative (not simultaneously zero) activation coefficients, the control gradient for contact i can be expressed as

$$\dot{\theta}_i = -k \left(\alpha_{FC} \frac{\partial \epsilon_{FC}}{\partial \theta_i} + \alpha_{MC} \frac{\partial \epsilon_{MC}}{\partial \theta_i} \right).$$

In the general case, the activation coefficients α will depend on the current context (state). The mapping $M : S \rightarrow \alpha$ from state to activation coefficients can be derived by knowledge-based heuristic rules or by reinforcement learning procedures (e.g., Q-learning).

• **Knowledge-based composition** schemes assume that control composition can be encoded by a collection of simple activation rules. Each activation coefficient is determined based on the relative significance of each controller to the task, measured in terms of the current control errors ($< \epsilon_{FC}, \epsilon_{MC} >$). In the grasp synthesis domain, a sensible heuristic rule is to address force residuals first, using the globally convex FC controller to position the contacts in the neighborhood of a solution, and then activate the locally convex MC controller to remove moment residuals.

• **Q-learning:** Starting with a random activation policy, the method will explore the whole state-action space and reward "good" solutions/penalize solutions corresponding to grasp failures. The mapping M that maximizes future rewards was derived after $45,000$ grasp trials, or $5,000$ trials per object (see [2]).

3 Control pre-imaging

Composite controllers may fail due to (1) interference between constituent controllers, (2) finite control repertoire, and (3) deficient composition functions. In principle, failure circumstances can be characterized by a succession of states in state space, and therefore one can predict the outcome of a sequence of control actions by learning to identify which states lead to failures or successes.

We define the success pre-image of a baseline controller as the union of all states from which the baseline control actions will ultimately generate successful solutions, as evaluated by a given success criterion. It is possible to compute the controller's success pre-image simply by propagating the convergent state's evaluation backwards in time. Every state *en route* to a successful solution belongs to the solution's success pre-image. Success prediction is a matter of recognizing which states belong to the success pre-image region. Once a predictor module is available, metalevel control actions (referred to as the pre-image control actions) can be invoked to steer the system back to the controller's success pre-image region whenever failure is predicted. This opportunistic activation of pre-image control actions enhances the performance of the baseline controller.

The use of predictable control actions to enhance system capabilities bears resemblance to Lozano-Pérez [10] approach to the automatic synthesis of fine-motion strategies, in which the final goal was propagated backwards through the actions of the underlying generalized impedance controller. In each pre-imaging step, the set of goal states increased until it includes the start state. When this happens, there exists a sequence of control actions that achieves the goal. The approach organizes the fine motion plan around the predictable behavior of the generalized damper control module. Notice however that in their work no estimation is involved because complete knowledge about the geometry and uncertainties is assumed; also, the associated planning procedure is completely off-line.

3.1 Implementation

Figure 1 depicts the block diagram for the pre-image (PI) grasp controller used in our experiments. The complete specification of a pre-image (PI) based controller involves the description of (1) a baseline controller, (2) a failure/success (F/S) predictor module, (3) the success criterion adopted, and (4) the pre-image control actions taken when failure is predicted.

• **Baseline controller:** The knowledge-based controller (Section 2.3) was used as the baseline controller.

• **F/S predictor:** The F/S predictor module was implemented as a neural network with two units for input of state information ($< \epsilon_{FC}, \epsilon_{MC} >$), three hidden units (one hidden layer) and one output unit (success expectation). Both the hidden and the output units were logistic units; the squashing function employed was the bipolar sigmoid function. Standard backpropagation was used to update the weights.

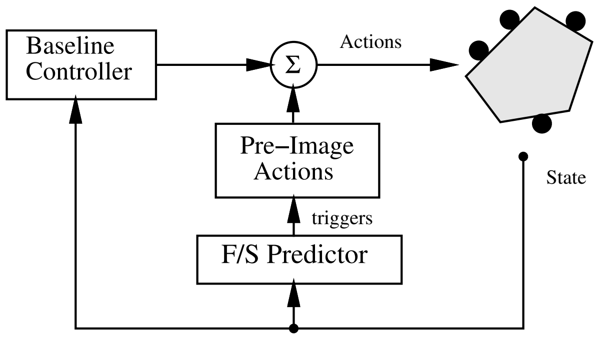


Figure 1: PI-based grasp controller.

• **Success criterion:** All convergent grasp configurations were scored according to the grasp evaluation metric, defined in [2]. This metric employs the magnitude of friction forces and the imbalance across the normal forces of each contact to classify convergent grasp configurations as satisfactory or not.

• **Pre-image actions:** There are two alternatives for the computation of corrective actions: (1) corrective actions can be designed for the specific problem, or (2) corrective actions can be computed using the predictor module itself. The predictor module effectively maps state to future reinforcement; if implemented by a neural network, the approximated forward model constructed can be used to compute how future reinforcement varies with respect to any state variables (as in [9]).

Our implementation employs the first option: once failure is predicted, the pre-image action adds a zero mean, Gaussian noise signal, bounded to $[-0.05, 0.05]$ radians (standard deviation $\sigma = 0.0167$), to the current control gradient. This small perturbation works efficiently to divert the system towards states upstream of successful grasp configurations.

3.2 Failure/Success prediction

The F/S predictor was trained over 10 epochs; each epoch consisted of 90 grasp trials, on 9 different objects, shown on Figure 2. The objects were presented in a random, fixed order for training. Ten grasp trials were executed for each object in the training sequence. A total of 900 grasp trials were attempted, or 100 trials for each object.

In each trial, four contacts are randomly placed on the object surface. The baseline controller executes

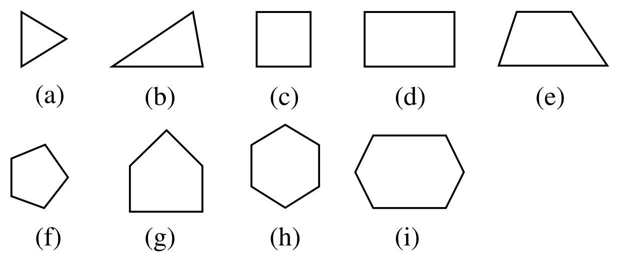


Figure 2: Objects used in our experiments: (a) Triangle, (b) Irreg. Triangle, (c) Square, (d) Rectangle, (e) Trapezoid, (f) Pentagon, (g) Irreg. Pentagon, (h) Hexagon, and (i) Irreg. Hexagon.

up to convergence[1], and the resulting grasp configuration is then labeled as either failure or as success, and this label is attached to all states visited in that trial. At the end of each epoch, visited states and attached labels are presented to the F/S predictor module as training instances.

4 Results

Figure 3 is the plot of average failure rate as a function of the number of training epochs, for the PI-based grasp controller (plot (a)) and the knowledge-based controller (plot(b), averaged over 500 epochs). For comparison purposes, the corresponding plot for the Q-learning based controller and is also shown (plot (c)). In all cases, an epoch consists of 90 grasp trials, or 10 grasp trials for each object in the object set. The F/S predictor module is used following the 10^{th} epoch to activate the pre-image control action. These plots were smoothed by averaging the raw data over a 10-epoch wide sliding window.

Figure 4(a) displays a typical path in state space, for the knowledge-based controller during the four contact grasp of a square object. Starting from the contact configuration shown in (c), the controller drives the system from the initial state in the right upper corner (corresponding to high force and moment closure errors) to a failure attractor (marked by an "X"). Figure 4(b) shows the corresponding path for the PI-based controller. Departing from the same initial configuration, the controller manages to converge to a success attractor (marked with "O"). Notice that object geometry affects the relation between ϵ_{FC} and ϵ_{MC}, affecting both the shape of the path

[1]Convergence is achieved if every contact has not moved more than 0.0005 radians during the last control step.

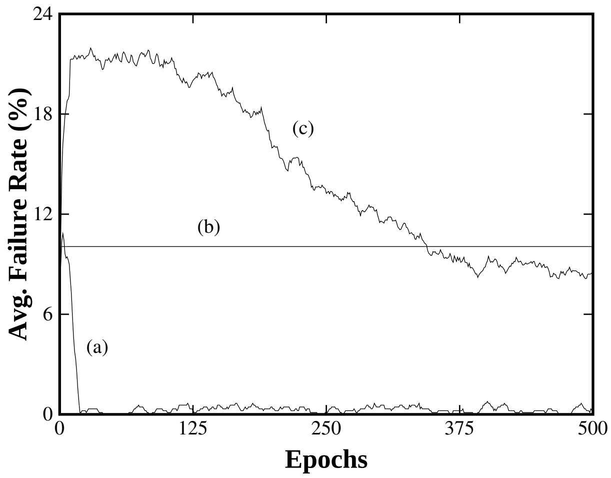


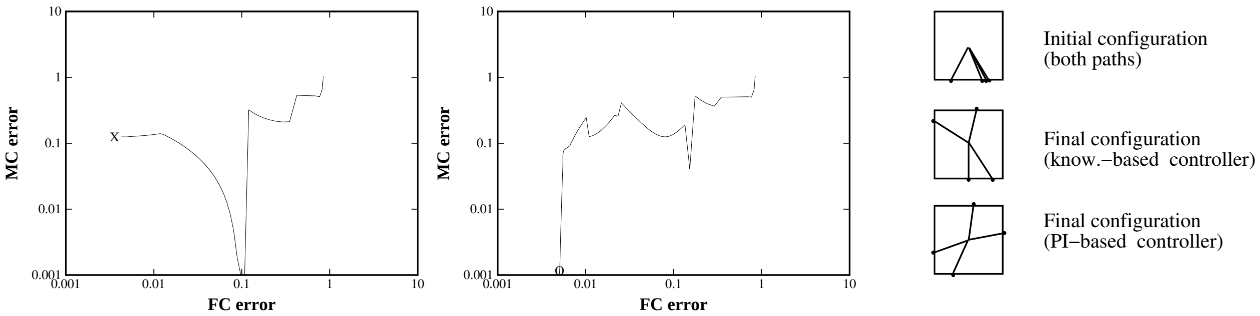
Figure 3: Average failure rate during training, for (a) PI-based controller, (b) knowledge-based controller, and (c) Q-learning based controller.

in state space and the position and number of success and failure attractors.

Figure 5(a) shows the output of the F/S predictor module (higher values denote greater failure expectancy) over the state space. As shown, failure expectancy is high for great values of ϵ_{MC}. Figure 5(b) shows the relative frequency each state in the state space is visited, for the knowledge-based controller attempting a four contact grasp of a square object. The two peaks correspond to the success and failure attractors; not surprisingly, the failure attractor is in the failure pre-image region. The final grasp configuration for both the failure and success attractors are also shown. Figure 5(c) shows the corresponding plot for the PI-based controller. Notice that only the peak corresponding to satisfactory solutions is present – no unsatisfactory solution was generated.

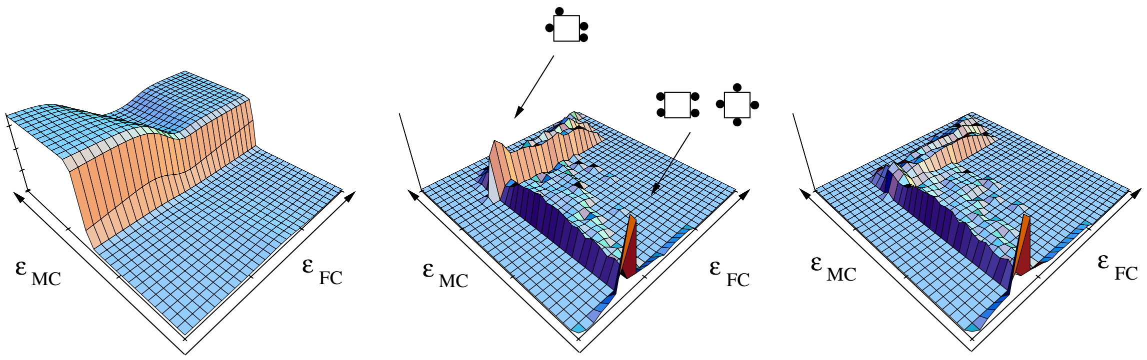
After the training stage, the performance of the PI-based controller was assessed for the four contact null grasp task of planar objects. Each object in the object set was submitted to 200 grasp trials. Table 1, column (c) reports the average percent failure rate observed for each object. For comparison purposes, the corresponding results for the knowledge-based controller and the Q-learning controller are reported in columns (a) and (b), respectively. The results in Table 1 show that the PI-based grasp controller performs remarkably well under the experimental conditions.

The grasp controller was used within the context of a pick-and-place task, where several control modules were integrated into a system capable of performing autonomous, collision-free reaching and grasping of wood blocks placed over a table at arbitrary positions. Figure 6 shows the experimental setup and a



(a) (b) (c)

Figure 4: Paths for (a) knowledge-based controller (converging to failure attractor), and (b) for the PI-based controller (converging to success attractor); (c) initial and final configurations, for both paths.



(a) (b) (c)

Figure 5: (a) Failure expectancy, as a function of state. Relative frequency each state is visited for (b) knowledge-based controller and (c) PI-based controller. The grasp configurations corresponding to the failure and the success attractors are also shown in (b).

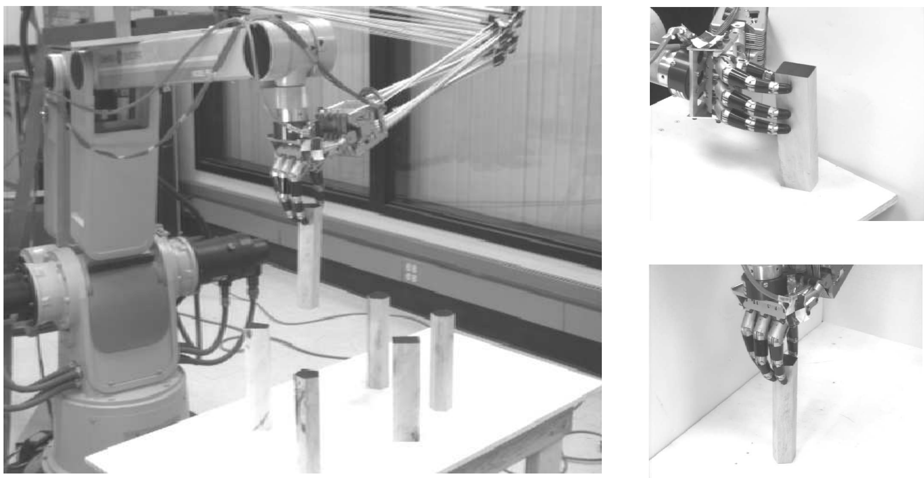


Figure 6: Experimental setup, showing the GE P50 robot arm and the Utah/MIT hand carrying an object with a top grasp. Side column shows the lateral grasp of a rectangular block and the top grasp of a pentagonal prism.

Table 1: Average failure rate per object, for different implementations and experiments.

Failure Rate (%)

Object	Know.-based control. (a)	QL (b)	PI-based control. (c)
Triangle	17.00	14.50	0.00
Square	20.50	9.00	0.00
Pentagon	0.00	0.00	0.00
Hexagon	0.00	0.00	0.00
Irreg.Triangle	20.50	22.00	2.50
Rectangle	12.00	3.50	0.00
Trapezoid	2.00	2.50	0.50
Irreg.Pentagon	18.50	23.00	0.00
Irreg.Hexagon	0.20	0.50	0.00
Average	10.06	8.33	0.33

top and lateral grasp synthesized by the controller.

5 Conclusion

The existence of formally derived, highly competent grasp controllers distinguishes our approach from other grasp synthesis approaches. These controllers are the generalization elements across the problem domain, providing for an adequate representation of the grasp state space. Furthermore, they structure and simplify the learning problem associated with control composition, through the use of control pre-imaging.

Control pre-imaging also provides a mechanism for resource allocation: corrective actions are activated only if the current context evokes a bad history of interaction controller-system. In principle, such corrective control actions may demand additional computational, kinematic, or sensory resources. The resulting resource allocation policy is context-dependent and efficient, committing new resources as they are required.

The unique aspect of our work is the synergy between control and learning: we have shown how learning augments an existing controller, and how the its control actions can be exploited to speed up the learning process. The results suggest that learning to identify successful interaction patterns between a predictable grasp controller and a class of object geometries is more efficient than learning a control policy from scratch (Q-learning). The use of prior system expertise explains the small number of grasp trials required in the construction of the F/S predictor module.

References

[1] Chen, I., and Burdick, J. Finding antipodal point grasps on irregularly shaped objects. In *Proc. 1992 IEEE Int. Conf. Robotics Automat.* (Nice, FRANCE, May 1992), vol. 3, pp. 2278–2283.

[2] Coelho Jr., J. Effective multifingered grasp synthesis. Master's thesis, UMass, Department of Computer Science, Amherst, MA, Sept. 1993.

[3] Coelho Jr., J., and Grupen, R. Optimal multifingered grasp synthesis. In *Proc. 1994 IEEE Int. Conf. Robotics Automat.* (San Diego, CA, May 1994).

[4] Faverjon, B., and Ponce, J. On computing two-finger force-closure graps of curved 2d objects. In *Proc. 1991 IEEE Int. Conf. Robotics Automat.* (Sacramento, CA, May 1991), vol. 1, pp. 424–429.

[5] Ferrari, C., and Canny, J. Planning optimal grasps. In *Proc. 1992 IEEE Int. Conf. Robotics Automat.* (Nice, FRANCE, May 1992), vol. 3, pp. 2290–2295.

[6] Grupen, R., Coelho Jr, J., and Souccar, K. On-line grasp estimator: A partioned state space approach. Tech. Rep. COINS Technical Report 92-75, COINS Department, University of Massachusetts, Oct. 1992.

[7] Guo, G., Gruver, W., and Jin, K. Grasp planning for multifingered robot hands. In *Proc. 1992 IEEE Int. Conf. Robotics Automat.* (Nice, FRANCE, May 1992), vol. 3, pp. 2284–2289.

[8] Jameson, J., and Leifer, L. Automatic grasping: An optimization approach. *IEEE Trans. Syst., Man, Cybern. SMC-17*, 5 (Sept./Oct. 1987).

[9] Jordan, M., and Rumelhart, D. Internal world models and supervised learning. In *Machine Learning: Proc. 8^{th} Int. Workshop* (1991), L. Birnhaum and G. Collins, Eds.

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

[11] Nguyen, V. Constructing stable grasps. *The Int. J. Robotics Res. 8*, 1 (1989), 26–37.