The 2009 IEEE/RSJ International Conference on
Intelligent Robots and Systems
October 11-15, 2009 St. Louis, USA

Robust sensor-based grasp primitive for a three-finger robot hand

Javier Felip and Antonio Morales

Abstract— **This paper addresses the problem of robot grasping in conditions of uncertainty. We propose a grasp controller that deals robustly with this uncertainty using feedback from different contact-based sensors. This controller assumes a description of grasp consisting of a primitive that only determines the initial configuration of the hand and the control law to be used.**

We exhaustively validate the controller by carrying out a large number of tests with different degrees of inaccuracy in the pose of the target objects and by comparing it with results of a naive grasp controller.

I. Introduction

Management of uncertainty is one of the biggest problem to address when developing applications for unstructured scenarios. In the case of robot grasping, uncertainty can arise from several sources: lack of complete knowledge about the physical properties and shape of the target objects, inaccuracy in the determination of the pose of the object and the configuration of the robot (i.e.: position of mobile robot), mismatch between planner models and real conditions due to sensing errors, limitation of planner models, and many others.

Analytical solutions to the grasp planning problem has been provided for structured scenarios [1]. However these solutions often depend on the assumption that the contact locations obtained as solutions are reachable by actuators with enough precision. For common robots scenarios this is not realistic even in the case that the shape of the object is perfectly known. Several attempts to design analytical grasp planning algorithms that take into account a certain degree of inacuracy has been made [2], [3]..

A common approach to reduce uncertainty is the use of sensor information in the planning and execution phases of grasping. Vision has been used to obtain the shape of unknown target objects [4], [5], or to determine the location and pose of them [6]. In both cases, visual input is used to plan feasible grasps. Visual feedback is also used when the arm tries to reach the object. Murphy et al. uses visual techniques to correct the orientation of four-finger hand while approaching an object to allow better contact locations [7]. Namiki et al. uses a fast control schema in combination with tactile feedback to cage an object [8]. Infrared sensors has been also used to correct the approaching orientation of the gripper [9]. In innovative design Hsiao et al. use IR sensors to estimate the normal direction of closer object surfaces to search a suitable contact location [10].

J. Felip and A. Morales are with Robotic Intelligence Laboratory at the Department of Computer Science and Engineering, Universitat Jaume I, 12006 Castellón, Spain {jfelip,morales}@uji.es

Fig. 1. PA-10 7 d.o.f with Barrett Hand and a JR3 force/torque and acceleration sensor. The hand has Weiss Robotics pressure sensors on its fingertips.

Once the object is contacted with the robot, gripper, tactile and force sensors can be applied. Contact force measurement is used to estimate the quality of the grasps [11], [12], [13] or the shape of the object [14] with the purpose of reach better contact locations through a sequence of grasping/regrasping actions. Contact information can also be used to program complex dexterous manipulation operations like finger repositioning while holding the object [11], [15]. Several works have combined the use of several sensors to complete the whole process of grasp planning and execution [16], [17].

Robustness in grasp execution is not only achieved by designing sensor-based controllers but also by combining several controllers with different optimisation goals. These combinations has been based on hierarchical schemes based on reflex programming [18], [13] or complementary controllers [12], [10].

A. Grasp primitives

In this paper we follow a sensor-based approach that is based on an alternative paradigm of describing grasps. Most of the above papers assume that grasps are described as a set of contact points on the object surface. In fact, most of the problems arise as a consequence of the impossibility of reaching those points. Our paper is developed under a different assumption. Grasps are described as instances of basic primitives [17]. A grasp primitive is a specific controller designed to perform a particular indivisible action, in our case a grasp. In practical terms it is defined by a initial hand preshape, a sensor-based controller, and a set of ending conditions. Its behaviour can be determined by several parameters like initial position and orientation, maximum force allowed, and others. An *instance* of a grasp primitive is the set of values of this parameters. Hence, a

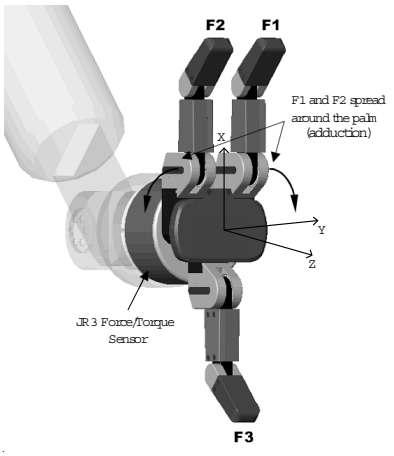

Fig. 2. Detail of the Barrett hand.

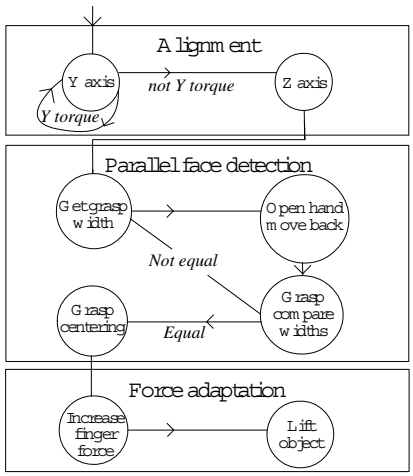

Fig. 3. Algorithm execution diagram.

grasp is an instance of a primitive that determines the initial configuration of the robot hand and the control policy that is going to be applied to execute the grasp.

This definition of grasp primitive presents two perspectives. From a practical point of view a grasp primitive is a single controller that performs a specific task on a particular embodiment. From an abstract point of view, primitives are the simplest pieces of a vocabulary to elaborate plans. Hence, they are well suited to be the basic piece of a reasoning and learning procedures.

The concept of grasp primitive is not new and has been used in many other robot-related works. Actually the term "motor primitive" is borrowed from neuroscience literature [19], and has also been widely used in robot learning [20], [21]. Nagatani and Yuta implemented and combined several action primitives to perform a complex behaviour: a mobile robot behaviour capable of opening and going through a door [22]. Aarno et al. implement visual analysis to program "Elementary Grasping Actions (EGA)", a kind of grasp primitives, for a parallel gripper [5]. Finally Finite State Machines has been proposed to combine primitive actions in the execution of complete manipulation tasks [23].

II. Methodology

A. System description and Assumptions

We implemented our primitive for a robotic setup consisting of a Mitsubishi PA-10 with 7 d.o.f. (Degrees of freedom) mounted on an Active Media PowerBot mobile robot. The manipulator is endowed with a three-fingered Barrett Hand and a JR3 force/torque and acceleration sensor mounted on the wrist, between the hand and the end-effector (see Fig. 1). The hand has been improved by adding on the fingertips arrays of pressure sensors designed and implemented by Weiss Robotics.

The Barrett hand is a 4 d.o.f., three-fingered hand. Each finger has one degree of freedom thus phalanxes are not independent. Fingers F1 and F2 can rotate around the palm and move next to Finger F3 (Thumb) or oppose to it, this d.o.f. is called adduction. The reference frame of the hand and the adduction d.o.f. are depicted in Fig. 2. Each finger of the hand has built-in strain sensor. The JR3 is a 12 d.o.f. sensor that measures force, torque and acceleration in each direction of the space.

Our experimental workspace consists of a horizontal surface where the targets objects are lying. The objects that can be manipulated are those that can fit inside the hand. The minimum dimensions are 25mm height, 70mm long, and 10mm width. Big objects that can be held by the hand should have a maximum width of 200mm. On this paper we have focused on box-like and clyinder-like objects within those dimensions (see Figs. 6, 7 and 8) and have not considered non-symetrical objects.

The input of the primitive controller is the starting position and orientation of the hand and the maximun finger force. In optimal conditions, the hand will be perfectly oriented in the direction of the object, and rotated perpendicularly with respect to the main axis of the object bounding box. These input parameters are provided from an external module based on visual information.

The designed controller is able to deal with errors in determining the parameters. The error estimitation is decomposed in translation error and rotation error (see Fig. 9). The first is defined by the cartesian distance on each frame axis between the center of the object and its estimation. The second is calculated from the difference between each estimated object axis and its true orientation.

The controller tries to grasp the object aproaching from above following an orientation close to the vertical. In order to allow lateral approaching directions several changes in the desing of the contorller may be necessary. Basically an estimation of the distance to the object should be known in advance. During the grasp execution, the robot can inadvertly move the object but the controller is designed to take into account most of these cases.

The algorithm starts with a cylindrical preshape. Where two fingers (F1 and F2) are opposing completely the thumb (F3). In some cases, the hand preshape is switched to a spherical configuration where the fingers are arranged

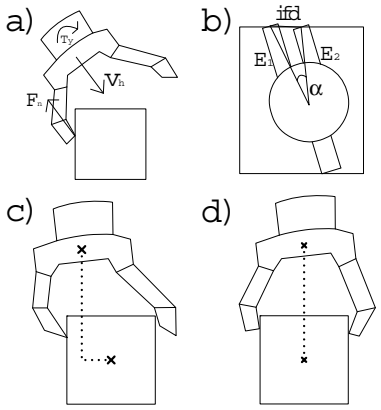

Fig. 4. Correction of alignment errors. a) The hand touches the object. The contact is not perpendicular and a normal force (Fn) appears at a distance from the center creating the torque Ty. b) The hand Z error is calculated using Index and Middle finger extensions. c) The grasp center is displaced.

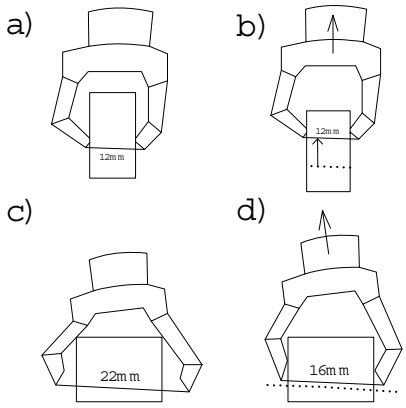

Fig. 5. Determining parallelism of grasped faces: a) First contact. b) Grasp width is constant, the faces grasped are parallel. c) Inner phanlanxes contact the object.

forming an equilateral triangle.

B. Algorithm

The controller tries to obtain grasp stability on the basis of the following criteria (ordered by relevance):

- Hand-object alignment
- Parallelism of contacted faces
- Maximization of contact surface
- Finger position symmetry
- Finger force symmetry

Hand and object are aligned when all their axis are parallel and the projection of the Z axis of the hand intersects the object bounding box on its center. The alignment avoids torque forces from appearing when lifting the object.

The starting position and orientation of the hand is a critical parameter for the algorithm. This pose sets the approach vector to the object which is along the positive Z axis of the hand (see Fig. 2). The aim of the algorithm is to perform a stable grasp of the object allowing a considerable error in the starting position. Thus the success of the grasp is less dependent from the accuracy of the estimations.

The execution of the grasp is divided into three main phases (see Fig. 3).

1) Alignment: This phase tries to align hand and object using force and tactile feedback. First of all the hand moves forward until the force/torque sensor on the wrist detects contact with the object. If this contact causes a torque force around Y axis it means that the object and the hand are not aligned (see Fig. 4.a). To correct this error the hand moves back and rotates an angle of two degrees, then continues touching the object until the torque disappears or it changes the sign. If the torque changes the sign, the hand rotates one degree in the opposite direction and the Y alignment ends.

At this point the hand closes. The difference in the extension of the fingers F1 and F2, determines the rotation around the Z axis (see Fig. 4.b) needed to align with the object. Both fingers must have the same extension to perform an stable grasp. This correction is not applied if the spherical pregrasp shape is set.

2) Parallel face detection: In the second stage the controller tries to determine if the contacted surfaces are parallel and stable. Using the Barrett Hand inner force sensors to stop the fingers and the hand propioception, the width of the current grasp is measured (see Fig. 5.a and c), then the hand opens a little and moves 5mm backwards. After that the hand closes and the width of the grasp is measured again. If there is a big difference between the two samples it means that the grasped faces are not parallel (see Fig. 5.d) and the process starts again. This phase of the algorithm repeats until the difference is close to zero or the object is lost. If the object is lost a reflex to recover it, is triggered. This reaction is explained at the end of this section.

When the grasp is stable(i.e. aligned with object and grasped surfaces parallel), the fingers move the object aligning it with the palm center and keeping the extension of the fingers (see Fig. 4.c and d) in order to improve contact surface, finger position symmetry and finger force symmetry.

3) Force adaptation: Using the fingertip integrated force sensors of the Barrett hand, the force of each fingertip is increased until it reaches the predefined limit. Then, the hand lifts the object and evaluates if the grasp has been successful.

C. Security reflexes

During the execution of the primitive, the algorithm is attentive for some important events in order to inform the user, adapt to the environment conditions and perform a successful grasp. The first event is the loss of the object. This happens when all the three fingers close completely without making any contact on the object. In this case the hand opens and moves a little bit down then closes again trying to recover the object contact.

Another event is the adduction of the fingers. When the fingers make contact, if the object is cylindrical, the fingers can adduct due to the adduction d.o.f is set free. This natural adduction is detected by the algorithm and the hand configuration is set to spherical.

The last event is the miss of the object by only one finger. The reaction is to open the hand and to move laterally 5cm (inter finger distance).

Fig. 6. Cylinder-like objects. Properties (radius x height, weight) from left to right and top to bottom: Cylinder1(110x80, light) Cylinder2(65x215, light) Cylinder3(105x75, heavy) Cylinder4(115x50, light)

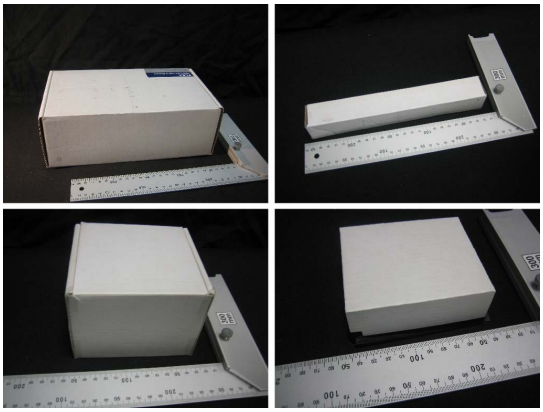

Fig. 7. Box-like objects. Properties (base x height, weight) from left to right and top to bottom: Box1(270x53x95, light) Box2(236x35x35, light) Box3(127x116x92, heavy) Box4(100x87x45, light)

D. Additional parameters

Other parameters as distance, size, weight and shape could be used to improve the accuracy and execution of the grasp. The distance could be used to avoid blind first contact with the object; the size could be used to set the starting opening of the fingers; the weight to determine the force to be applied by the fingers; and the shape to set the pregrasp shape reducing the time consumed by the pregasp shape detection and switching.

III. Validation

A test bench has been designed in order to validate the grasping controller. This test bench consists of a set of objects and a set of starting positions to be tried with each object.

To have a comparison reference for our controller, we have designed an alternative naive grasp controller without corrections. This controller needs 4 input parameters: starting position, distance to the object, pregrasp size and finger force. The fingers moves to the pregrasp size and the hand moves forward along its Z axis the distance specified. The hand closes and lifts the object. If the object is lifted and does not fall for 10 seconds, the execution is successful.

A. Objects and test bench

The objects selected are classified according to their shape (cylinders in Fig. 6, boxes in Fig. 7 and others in Fig. 8), their size (thin, normal, thick) or their weight (light, heavy). All the objects are solid. Following the shape classification, we have selected thin, normal and thick objects for each shape in order to test as many different combinations of object features as possible. The optimal conditions have been tested in all the objects. We have selected a subset of 2 box-like objects and 2 cylinder-like objects to test rotation and translation error conditions. This selected objects are the biggest and the smallest from each category.

Translation error is the deviation from the center of the object to the center of the hand and it is measured in mm. Rotation error is the deviation between hand and object main axis and is measured in degrees (see Fig. 9).

This test bench evaluates robustness against translation and rotation errors. The behaviour of each algorithm has been also evaluated in optimal conditions which can present a rotation error of 5 degrees and 10% of translation error.

To evaluate the effect of rotation errors the following conditions have been taken into consideration:

- 15 degrees on X, Y, Z, XY, XZ, YZ and XYZ.
- 20 degrees on X, Y, Z

The combined rotation error is applied first in X next in Y and later in Z. A rotation error of 15 degrees in XYZ is a rotation of 15 degree on X, then 15 degree on Y and finally 15 degree on Z. The results of the rotation tests are shown in Table III for the robust controller and in Table IV for the naive controller. The cylinder-like objects are invariant to rotation in Z axis. The Z rotation error is not applicable to cylinder-like objects.

The amount of translation error is relative to the size of the object because usually this two variables (size and error) are related. To evaluate the effect of translation errors the following conditions have been taken into consideration:

- 20% on X, Y, Z, XY, XZ, YZ and XYZ
- 40% on X, Y, Z

IV. Results

The global results are presented in Table I and Table II, the first column shows the results for the optimal case, the

Fig. 8. Other objects. Properties (base x height, weight) from left to right: Other1(180x90x90, heavy) Other2(90x90x163, light)

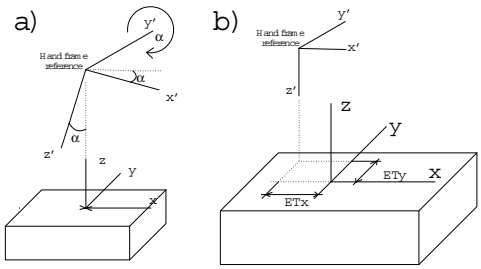

Fig. 9. Rotation and translation error, the object frame reference is on the center of the object. a) Example of alpha degrees Y rotation error. b) Example of X and Y translation error

	Optimal	Rotation	Translation	Total
Box 1	5/5(100%)	24/24(100%)	13/24(54%)	42/53(79%)
Box 2	5/5(100%)	16/22(73%)	22/24(92%)	43/51(84%)
Cylinder 2	5/5(100%)	7/12(58%)	16/20(80%)	28/37(76%)
Cylinder 4	5/5(100%)	11/11(100%)	19/20(95%)	35/36(97%)

TABLE I

ROBUST ALGORITHM GLOBAL RESULTS

second and third columns present the summary of the rotation and translation error. The last column shows the averaged results for each object.

Details about experiments with error conditions can be found in Table III and Table IV for rotation error and in Table V and Table VI for translation error.

V. DISCUSSION

Summary tables I and II show clearly the better performance obtained by our robust controller in comparison with the naive one. It not only successes in a 100% of the experiments in optimal conditions but also outperforms

	Optimal	Rotation	Translation	Total
Box 1	5/5(100%)	25/50(50%)	16/40(40%)	46/95(48%)
Box 2	5/5(100%)	26/50(52%)	40/40(100%)	71/95(75%)
Cylinder 2	5/5(100%)	23/25(92%)	35/40(88%)	63/70(90%)
Cylinder 4	5/5(100%)	24/25(96%)	14/20(70%)	43/50(86%)

TABLE II

NAIVE ALGORITHM GLOBAL RESULTS

Error	Box 1	Box 2	Cylinder 2	Cylinder 4
15 X Axis	2/2(100%)	2/2(100%)	2/2(100%)	2/2(100%)
20 X Axis	2/2(100%)	2/2(100%)	2/2(100%)	2/2(100%)
15 Y Axis	3/3(100%)	2/2(100%)	1/2(50%)	3/3(100%)
20 Y Axis	3/3(100%)	2/2(100%)	1/4(25%)	2/2(100%)
15 Z Axis	3/3(100%)	2/2(100%)	N/A	N/A
20 Z Axis	3/3(100%)	2/2(100%)	N/A	N/A
15 XY Axis	2/2(100%)	1/2(50%)	2/2(100%)	2/2(100%)
15 XZ Axis	2/2(100%)	1/2(50%)	N/A	N/A
15 YZ Axis	2/2(100%)	1/2(50%)	N/A	N/A
15 XYZ Axis	2/2(100%)	1/4(25%)	N/A	N/A

TABLE III

RESULTS WITH ROTATION ERROR FOR THE ROBUST ALGORITHM

Error	Box 1	Box 2	Cylinder 2	Cylinder 4
15 X Axis	5/5(100%)	5/5(100%)	5/5(100%)	5/5(100%)
20 X Axis	5/5(100%)	5/5(100%)	5/5(100%)	5/5(100%)
15 Y Axis	4/5(80%)	2/5(20%)	3/5(20%)	4/5(80%)
20 Y Axis	2/5(20%)	0/5(0%)	5/5(100%)	5/5(100%)
15 Z Axis	4/5(80%)	5/5(100%)	N/A	N/A
20 Z Axis	4/5(80%)	5/5(100%)	N/A	N/A
15 XY Axis	0/5(0%)	0/5(0%)	5/5(100%)	5/5(100%)
15 XZ Axis	0/5(0%)	4/5(80%)	N/A	N/A
15 YZ Axis	1/5(20%)	0/5(0%)	N/A	N/A
15 XYZ Axis	0/5(0%)	0/5(0%)	N/A	N/A

TABLE IV

RESULTS WITH ROTATION ERROR FOR THE NAIVE ALGORITHM

Error	Box 1	Box 2	Cylinder 2	Cylinder 4
20% X Axis	2/2(100%)	2/2(100%)	2/2(100%)	2/2(100%)
40% X Axis	1/2(50%)	2/2(100%)	2/2(100%)	2/2(100%)
20% Y Axis	1/2(50%)	2/2(100%)	1/2(50%)	2/2(100%)
40% Y Axis	0/2(0%)	2/2(100%)	0/2(0%)	1/2(50%)
20% Z Axis	2/2(100%)	2/2(100%)	2/2(100%)	2/2(100%)
40% Z Axis	2/2(100%)	2/2(100%)	2/2(100%)	2/2(100%)
20% XY Axis	1/4(25%)	3/4(75%)	2/2(100%)	2/2(100%)
20% XZ Axis	2/2(100%)	1/2(50%)	2/2(100%)	2/2(100%)
20% YZ Axis	1/2(50%)	1/2(50%)	1/2(50%)	2/2(100%)
20% XYZ Axis	1/4(25%)	3/4(75%)	2/2(100%)	2/2(100%)

TABLE V

RESULTS WITH TRANSLATION ERROR FOR THE ROBUST ALGORITHM

the naive one when rotational and translational errors are introduced.

The only exemption to this rule is the case of Cylinder 2 (object on the top-left corner on fig 6). This object is too light and when touched while lying on a surface, it moves easily. We observed that the successive contacts that our controller produce causes that the object variates its position making impossible to grasp it.

This case shows one of the drawbacks of our approach. Our controller is touching the objects several times before finally closing the finger to catch them. In case of light or unstable objects this can be a problem. This difficultty surpassed by the use of more sensitive sensors or by the implmentation of compliant hardware or controllers. The use of proximitiy sensors [10] would completely solve this

Error	Box 1	Box 2	Cylinder 2	Cylinder 4
20% X Axis	0/4(0%)	4/4(100%)	4/4(100%)	4/4(100%)
40% X Axis	0/4(0%)	4/4(100%)	3/4(75%)	4/4(100%)
20% Y Axis	4/4(100%)	4/4(100%)	4/4(100%)	4/4(100%)
40% Y Axis	0/4(0%)	4/4(100%)	4/4(100%)	0/4(0%)
20% Z Axis	3/4(75%)	4/4(100%)	4/4(100%)	2/4(50%)
40% Z Axis	2/4(50%)	4/4(100%)	2/4(50%)	2/4(50%)
20% XY Axis	0/4(0%)	4/4(100%)	4/4(100%)	4/4(100%)
20% XZ Axis	2/4(50%)	4/4(100%)	4/4(100%)	2/4(50%)
20% YZ Axis	4/4(100%)	4/4(100%)	2/4(50%)	4/4(100%)
20% XYZ Axis	1/4(25%)	4/4(100%)	4/4(100%)	2/4(50%)

TABLE VI

RESULTS WITH TRANSLATION ERROR FOR THE NAIVE ALGORITHM

problem.

One of the advantages of our approach is the little previous information it needs about the object. No exact model of the object is necessary, and the only input is the maximum force to be applied by the fingers. It is supposed that the hand is appropriately oriented and preshaped. More information, like estimated size or the distance, would be definitively help to improve the controller robustness and the time necessary to complete a grasp.

Currently all the grasp tried to approach from above. That is, the objects are lying on a surface and the hand approaches them vertically. This simplifies our controller since the movements of the objects are limited. Improvements are necessary if grasps from a side are going to be executed, since the stability of the objects could be compromised if they are touch. In this case an estimation of the distance to the object would be necessary.

At the moment the average time to execute a grasp is about 40 seconds, though this time depends on the object and the initial position error. It could be reduced providing more information about the location and characteristics of the objects.

Finally, an attached video shows pose correction phases, event adaptation and grasp force increasing. It is also shown that the stability of the grasps performed by the robust algorithm are better than the ones performed by the naive controller.

VI. Conclusion

We have developed a robust sensor-based grasp primitive that need little information to execute its task and that is able the correct and adapt to variations and inaccuracies in the expected conditions of the scenario.

We indicated three ways of improving the grasp primitive controller implemented. The most immediate future work is to extend the family of manipualtion primitives that allow the execution of a complete pick-and-place task, i.e.: approaching and preshaping, lifting, transportation and landing primitives. The development of these primitives would provide a vocabulary of basic skills that will allow planning, and learning in future stages.

Acknowledgment

This paper describes research carried out at the Robotic Intelligence Laboratory of Universitat Jaume I. The research leading to these results has received funding from the European Community's Seventh Framework Programme FP7/2007-2013 under grant agreement 215821 (GRASP project), and by Fundació Caixa-Castelló (P1-1A2006-11).

References

[1] A. Bicchi, "Hand for dexterous manipulation and robust grasping: A difficult road towards simplicity," *IEEE Transactions on Robotics and Automation*, vol. 16, no. 6, pp. 652–662, 2000.

[2] R. C. Brost, "Planning robot grasping motions in the presence of uncertainty," Tech. Rep. CMU-RI-TR-85-12, Computer Science Department and Robotics Institute, Carnegie Mellon University, 1985.

[3] Y. Zheng and W.-H. Qian, "Coping with the grasping uncertainties in force-closure analysis," *International Journal of Robotics Research*, vol. 24, no. 4, pp. 311–327, 2005.

[4] A. Morales, P. Sanz, A. del Pobil, and A. Fagg, "Vision-based three-finger grasp synthesis constrained by hand geometry," *Robotics and Autonomus Systems*, vol. 54, pp. 496–512, June 2006.

[5] D. Aarno, J. Sommerfeld, D. Kragic, N. Pugeault, S. Kalkan, F. Wörgötter, D. Kraft, and N. Krüger, "Early reactive grasping with second order 3D feature relations," in *IEEE Conference on Robotics and Automation (submitted*, 2007.

[6] P. Azad, T. Asfour, and R. Dillmann, "Combining appearance-based and model-based methods for real-time object recognition and 6D-localization," in *International Conference on Intelligent Robots and Systems (IROS)*, (Beijing, China), 2006.

[7] T. Murphy, D. Lyons, and A. Hendriks, "Stable grasping with a multi-fingered robot hand: A behavior-based approach," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, vol. 2, (Yokohama, Japan), pp. 867–874, July 1993.

[8] A. Namiki and M. Ishikawa, "Optimal grasping using visual and tactile feedback," in *IEEE/SICE/RSJ Intl. Conf. on Multisensor Fusion and Integration for Intelligent Systems*, (Whashington, DC), pp. 589–596, Dec. 1996.

[9] M. Teichmann and B. Mishra, "Reactive robotics I: Reactive grasping with a modified gripper and multi-fingererd hands," *International Journal of Robotics Research*, vol. 19, no. 7, pp. 697–708, 2000.

[10] K. Hsiao, P. Nangeroni, M. Huber, A. Saxena, and A. Y. Ng, "Reactive grasping using optical proximity sensors," in *IEEE International Conference on Robotics and Automation*, (Kobe, Japan), May 2009. To appear.

[11] J. Coelho Jr. and R. Grupen, "A Control Basis for Learning Multifingered Grasps," *Journal of Robotic Systems*, vol. 14, pp. 545–557, Oct. 1997.

[12] R. Platt Jr., A. H. Fagg, and R. Gruppen, "Nullspace composition of control laws for grasping," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, (Lausanne, Switzerland), pp. 1717–1723, 2002.

[13] T. Mouri, H. Kawasaki, and S. Ito, "Unknown object grasping strategy imitating human grasping reflex for anthropomorphic robot hand," *Journal of Advanced Mechanical Design, Systems, and Manufacturing*, vol. 1, no. 1, pp. 1–11, 2007.

[14] P. Allen and K. Roberts, "Haptic object recognition using a multi-fingered dextrous hand," *Robotics and Automation, 1989. Proceedings., 1989 IEEE International Conference on*, pp. 342–347 vol.1, May 1989.

[15] M. Huber and R. Grupen, "Robust finger gaits from closed-loop controllers," *IEEE/RSJ International Conference on Intelligent Robots and Systems*, vol. 2, pp. 1578–1584 vol.2, 2002.

[16] P. Allen, A. T. Miller, P. Oh, and B. Leibowitz, "Using tactile and visual sensing with a robotic hand," in *IEEE International Conference on Robotics and Automation*, (Albuquerque, New Mexico), pp. 677–681, Apr. 1997.

[17] B. J. Grzyb, E. Chinellato, A. Morales, , and A. P. del Pobil, "Robust grasping of 3D objects with stereo vision and tactile feedback," in *International Conference on Climbing and Walking Robots and the Support Technologies for Mobile Machines (CLAWAR)*, (Coimbra, Portugal), pp. 851 – 858, 2008.

[18] K. Imazeki and T. Maeno, "Hierarchical control method for manipulating/grasping tasks using multi-fingered robot hand," *Intelligent Robots and Systems, 2003. (IROS 2003). Proceedings. 2003 IEEE/RSJ International Conference on*, vol. 4, pp. 3686–3691 vol.3, Oct. 2003.

[19] F. A. Mussa-Ivaldi, S. F. Giszter, and E. Bizzi, "Linear combinations of primitives in vertebrate motor control," *Proceedings of the National Academy of Sciences of the United States of America*, vol. 91, no. 16, pp. 7534–7538, 1994.

[20] M. Ferch and J. Zhang, "Learning cooperative grasping with the graph representation of a state-action space," *Robotics and Autonomous Systems*, vol. 38, pp. 183–195, 2002.

[21] R. Amit and M. J. Mataric, "Parametric primitives for motor representation and control," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA-2002*, pp. 863–868, 2002.

[22] K. Nagatani and S. Yuta, "Door-opening behavior of an autonomous mobile manipulator by sequence of action primitives," *Journal of Robotic Systems*, vol. 13, no. 11, pp. 709–721, 1996.

[23] D. Kragic, S. Crinier, D. Brunn, and H. Christensen, "Vision and tactile sensing for real world tasks," *IEEE International Conference on Robotics and Automation*, pp. 1545–1550, September 2003.