2007 IEEE International Conference on
Robotics and Automation
Roma, Italy, 10-14 April 2007

Task-Oriented Grasping using Hand Preshapes and Task Frames

Mario Prats, Pedro J. Sanz, Angel P. del Pobil
Robotic Intelligence Lab
Universitat Jaume I
Castellón, Spain
[mprats,sanzp,pobil]@icc.uji.es

Abstract— **In this paper we present a robot that is able to perform daily manipulation tasks in a home environment, such as opening doors and drawers. Taking as input a simplified object model and the task to perform, the robot automatically finds a grasp suitable for the task and performs it. For this, we identify a set of hand preshapes and classify them according to the grasp wrench space they generate. Given a task, the robot selects the most suitable hand preshape and automatically plans a set of actions in order to reach the object and to perform the task, taking continuously into account the task forces. The concept of hand preshape is extended for the inclusion of a task frame, which is a concept from task planning, thus filling the gap between the grasp and the task.**

I. Introduction

Manipulation is a well-known problem in intelligent robotics [1]. In the community there has been a focus on grasping [2] [3], however, physical interaction through manipulation in our daily life even for simple and common tasks goes well beyond grasping for picking and placing. Examples of those tasks are: switching on the light, taking a book out of a bookshelf, opening a door or a drawer, pushing objects, turning on a tap, etc. These are basic skills that need to be incorporated into future service robots in a robust and flexible way.

When interacting with an object, task is important. There are few works about grasping that take the task into account. The first was presented by Li and Sastry [4]. In their work, they defined three grasp quality measures, one of which was task-oriented. More recent works, as [5] [6], try to include the task into the grasp evaluation, as a quality measure. They consider that a grasp has already been found, and evaluate the suitability of the given grasp for the desired task. In practice, lots of grasps would have to be generated and evaluated, making these approaches computationally unaffordable. The task should be taken into account from the early planning stages: instead of a quality measure, it is rather needed as a heuristic for grasp planning.

Our approach considers the task from the beginning. Taking as input a simplified object model and the task to perform, the intelligent robot automatically finds a grasp suitable for the task and performs it. We make use of hand preshapes, which have already been successfully applied to pick and place grasping operations [3] [7]. The most common hand preshapes are considered and classified according to the grasp wrench space they can generate [5]. When the intelligent robot is requested to perform a task, it selects the most suited hand preshape, and then it plans the grasp on the object. Then, the task is performed taking into account the task forces. The bridge between grasp planning and task planning is the task frame [8], which is defined for each hand preshape, thus making the concept of hand preshape suitable for practical task-oriented grasping. As far as we know, this is the first work that puts into practice task-oriented grasping. Our aim is to define a multipurpose service robot, able to deal not only with one task, but with several home daily tasks.

In Section II a set of task-oriented hand preshapes for the Barrett Hand [9] are defined. Section III shows how the concept of task frame [8] can be used for task-oriented grasping. In Section IV a task-oriented grasp planning algorithm is presented, whereas Section V is devoted to the task planning algorithm. Finally, a practical example is explained in Section VI, and some conclusions and future lines are outlined in Section VII.

II. Task-oriented hand preshapes

The use of hand preshapes for robotic grasping was first proposed by [7], and satisfactory results have been obtained since then [10] [3]. A hand preshape is a predefined prototype of grasp. Instead of adapting hand kinematics to a given grasp, we approach the grasp to a grasp prototype, which we know how to perform with our hand. There are evidences that humans use hand preshapes for grasping [11].

Until now, hand preshapes have been used for the widely considered pick and place task [7] [3]. No work has considered hand preshapes for performing other types of tasks such as pushing or turning objects. In our work, we identify a set of four task-oriented hand preshapes for the Barrett Hand [9]:

- **Hook power**. Fingers are set into a hook configuration. There is no opposing thumb, but the hook encloses the object, thus making a possible contact with the distal phalanx, the proximal phalanx and the palm. This hand configuration is ideal for grasping handles and pushing along a known direction.
- **Hook precision**. Fingers are set into a hook configuration. There is no opposing thumb, and contact is made with the fingertips. This preshape is suitable for pushing objects when the hook power configuration is not possible, as for example, when there is no space behind the object for placing the distal phalanx.

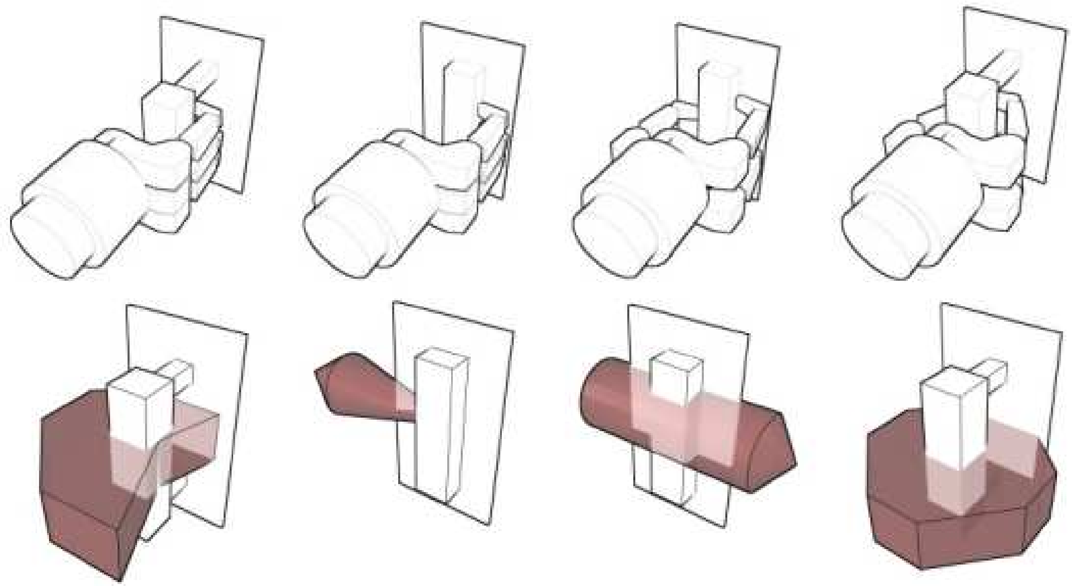

Fig. 1. Upper row: task-oriented hand preshapes. Lower row: simplified representation of the GWS for each preshape. From left to right: hook power, hook precision, precision and cylindrical.

- **Precision**. There is a thumb that opposes the other fingers. Thus, force can be exerted along the two senses of a same direction. Contact is made with the fingertips. This grasp is useful when the direction of the motion is known, but not the sense. It can also be used for exerting a torque, as when turning on a tap.
- **Cylindrical** As in the previous case, the thumb opposes the other fingers. However, with this configuration the fingers enclose the object and make force towards the palm. This is a firm grasp that can be used when there is enough space around the object and the direction of motion is unknown.

Figure 1 depicts the four task-oriented hand preshapes that have been identified for the Barrett Hand, along with an approximation of the grasp wrench space (GWS) [5] for each preshape, considering only the translational degrees of freedom and point contacts with friction. The GWS [5] is the set of all possible wrenches that can be applied on the object through the contacts. From the drawings of Figure 1, it can be seen that the most effective task-oriented preshape is the cylindrical one. With this preshape, forces can be applied in all directions, even in the vertical direction if the contacts have enough friction. However, it is not always possible to enclose the object with the fingers. In those cases, a precision preshape can be used. But if we know a priori the force vector that we want to apply, the grasp can be improved by using either the hook power or the hook precision preshapes. In these two configurations, all fingers point in the same direction and sense, maximizing the net force for the task. In both cases, certain directions of the GWS are narrowed while others are enlarged. The hook power preshape can be used when the task is well known and there is space for enclosing the object. If there is no space around the object, the hook precision preshape will be selected.

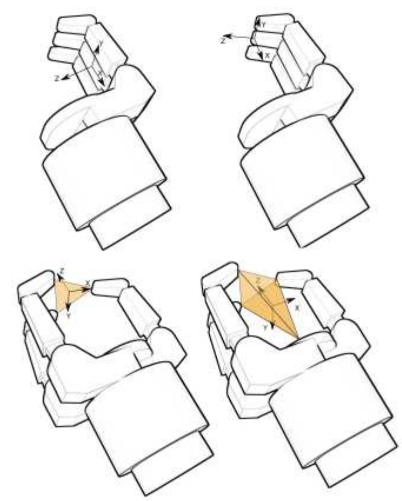

Fig. 2. Task frames for the hook power (top-left), hook precision (top-right), precision (bottom-left) and cylindrical (bottom-right) preshapes.

III. TASK FRAMES AND HAND PRESHAPES

The concept of task frame (TF) was introduced by Mason [8] under the Task Frame Formalism (TFF). It is a powerful concept that has been widely used for compliant task execution [12] [13] [14] [15]. Within this framework, the whole space of possible task directions is divided into two orthogonal subspaces: one composed of force-controlled directions (position-constrained), and the other that represents velocity-controlled directions (force-constrained). A great number of tasks can be performed within this framework. The only condition is that it must be possible to decompose the task into force and velocity-controlled directions [16] [15].

In the same manner that grasp planning approaches usually do not take into account the task, task planning works do not take into account the grasp. They usually consider that a suitable grasp has been performed and focus on the compliant task execution part. There is a gap between grasp planning and task planning that needs to be filled. We propose to associate a task frame to each hand preshape, so that a task can be easily planned in terms of the grasp. Whereas the hand preshape tells the robot how to configure its hand, the task frame serves as a reference system where to specify the hand motion. The goal of the robot will be to align the hand's task frame with an object's frame that will be selected during task planning.

Figure 2 shows the task frame that has been selected for each of the hand preshapes. For the hook power preshape, the TF has been chosen to be at the proximal phalanx with Z direction being normal to the finger's surface. This is the point that will be controlled for making contact with the object. For the hook precision configuration, contact must be made with the fingertips, and for that reason the TF has been placed there. When grasping an object with the precision preshape, the TF is chosen to be at the centroid of the triangle composed by the three fingertips, with Z direction normal to the palm surface. Finally, for the cylindrical preshape, the TF is placed at the center of mass of the pyramid that has the three fingertips as base and the center of the palm as the vertex.

As it will be seen in next sections, the inclusion of the task frame in the hand preshapes will help both the grasp and the task. In our approach the task frame is the common link between grasp planning and task planning.

IV. Task-oriented grasp planning

The goal of task-oriented grasp planning is to solve the following question: given an object and a task, how can I grasp the object to efficiently perform the task?. To solve this problem, the robot needs a suitable object representation, including the potential tasks that can be performed with it. In section IV-A an object representation, suitable for describing home daily tasks is proposed. Section IV-B describes a task-oriented grasp planning algorithm that takes the object representation as input for computing the task-oriented grasp.

A. Object representation

Grasp planning algorithms usually take as input an object model [3] [17] [18], that can be acquired by different techniques as 3D reconstruction, object recognition, etc. Some approaches consider a detailed 3D model [18], whereas others simplify the problem by using shape primitives [3]. We think that a detailed 3D model is not necessary for planning a grasp. Grasps computed in this manner usually do not take the fingers reachability restrictions into account, and end up with a set of contacts that cannot be reached with current robotic hands. On the other hand, approximating the object model by shape primitives results in faster and practical planning algorithms. Moreover, shape primitives fit very well with the concept of hand preshapes [3].

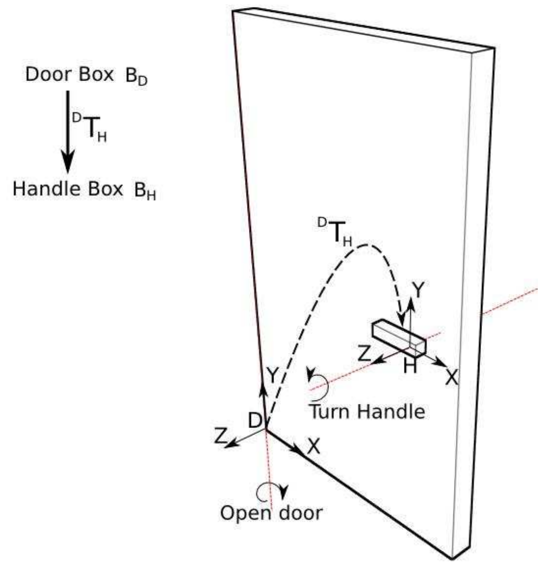

Fig. 3. A simplified model of a door.

We are interested in home tasks such as opening doors, drawers, windows, etc. As the shape of these objects is usually simple, we make use of a simple shape primitive: a box. Each object in our object database is modelled as a hierarchy of boxes, which can be viewed as a tree where each node corresponds to a box. Each node can have several children, but only one parent. The goal is to recursively define an object as the union of several (possibly articulated) subobjects. Figure 3 shows the simplified model of a door. There is a base box corresponding to the door, which has the handle box as a child. Each box is defined in its own reference system. The door box in Figure 3 is defined in frame $\mathcal{D}$, whereas the handle box is defined in the reference system $\mathcal{H}$. The pose of $\mathcal{H}$ with respect to $\mathcal{D}$ is also stored as an homogeneous transformation matrix ${}^{\mathcal{D}}T_{\mathcal{H}}$.

Apart from the simplified object model, a way of specifying the task is needed. In our approach, for each box of the object model, its degrees of freedom on the local reference system are stored. In the example of Figure 3, the door box has one degree of freedom (DOF): a rotation around Y axis. In the same manner, the handle box has another DOF around Z axis. Thus, the robot not only has a geometrical model of the object, but also a structural and mechanical model. It knows what can be done with the object. The task-oriented grasp planning will show the robot how to do it.

B. Task-oriented grasp planning algorithm

The task-oriented grasp planning algorithm takes as input the object model and the node, in the tree hierarchy, that contains the DOF to activate. In the example of Figure 3, the model is composed by the two boxes and their relationship. For turning the handle, the node is the handle box, and the DOF is the rotation about Z axis. The output of the algorithm is a target frame expressed with respect to any of the object frames, as well as a hand preshape suitable for the task. During the execution, the robot will try to move the task frame (i.e. its hand) towards the target frame. Being $B_{\mathcal{I}}$ the object that contains the DOF to activate on frame $\mathcal{I}$, the following steps are executed:

- **Search for handles.** The size of the boxes that are under $B_{\mathcal{I}}$ in the hierarchy is analyzed with the aim to find a handle-like box (one side much larger than the other two). This box, called $B_{\mathcal{J}}$, will be grasped by the robot for performing the task.
- **Transforming the task.** If the DOF on $\mathcal{I}$ is translational, it can be expressed as a three-component vector $\mathbf{t}_{\mathcal{I}}$, where the index of the corresponding DOF is set to 1. If the DOF on $\mathcal{I}$ is rotational, it is transformed into a translational DOF in the following manner: we search in $B_{\mathcal{I}}$ for the faces perpendicular to the rotational DOF (a total of four), and select the largest one. Being $\mathbf{n}$ the normal to the selected face, we set $\mathbf{t}_{\mathcal{I}} = -\mathbf{n}$. Finally, the tree-component vector $\mathbf{t}_{\mathcal{I}}$ is transformed into frame $\mathcal{J}$ according to the relationship $\mathbf{t}_{\mathcal{J}} = {}^{\mathcal{I}}R_{\mathcal{J}}^{-1}\mathbf{t}_{\mathcal{I}}$, where ${}^{\mathcal{I}}R_{\mathcal{J}}$ is the rotation matrix that aligns $\mathcal{J}$ with $\mathcal{I}$.
- **Selecting the grasp.** At this point we have the subobject to grasp $B_{\mathcal{J}}$ and a task direction $\mathbf{t}_{\mathcal{J}}$. The hand preshape and the target frame are selected according to the following classification (see Figure 4):
  1) The line with direction $\mathbf{t}_{\mathcal{J}}$ intersects with the parent of $B_{\mathcal{J}}$. It means that we need to make force towards the front or towards the back, such as closing or opening a drawer. If there is enough space for enclosing $B_{\mathcal{J}}$, a hook power grasp will be selected. The target frame will be on the center of a face with normal perpendicular to $\mathbf{t}_{\mathcal{J}}$. If there is no enough space, the precision preshape will be chosen, and the target frame will be in the center of mass of the box $B_{\mathcal{J}}$.
  2) The line with direction $\mathbf{t}_{\mathcal{J}}$ is parallel to the parent of $B_{\mathcal{J}}$ and the sense of the task is known. The task is to push towards a known direction and sense, as for example, opening a sliding door to the left. The force vector is well-known. So, a hook power grasp will be selected if the object can be enclosed, and a hook precision grasp otherwise. The face with normal equal to the force vector is found, and the target frame is set to the middle point of this face.
  3) The line with direction $\mathbf{t}_{\mathcal{J}}$ is parallel to the parent of $B_{\mathcal{J}}$ and the sense of the task is unknown. The task is to push towards the left/right or top/down, such as when opening/closing a sliding door. As the sense is unknown, we should be able to make force in both senses. Thus, a cylindrical grasp is chosen if the object can be enclosed, and a precision grasp otherwise. In both cases, the target frame is placed at the center of mass of the box.

Figure 4 shows a typical example for each case. The first situation is illustrated with the example of a drawer, whereas the other two are exemplified with a sliding door. The frame denoted as $\mathcal{T}$ is the target frame, which is the output of the grasp planning algorithm, along with the most suitable hand preshape. As we have defined a task frame for each preshape (Figure 2), the robot can easily perform the grasp by making the task frame and the target frame to coincide. Moreover, task planning will take the task frame as reference. Therefore, the task frame is used to perform both the grasp and the task, thus serving as bridge between grasp planning and task planning.

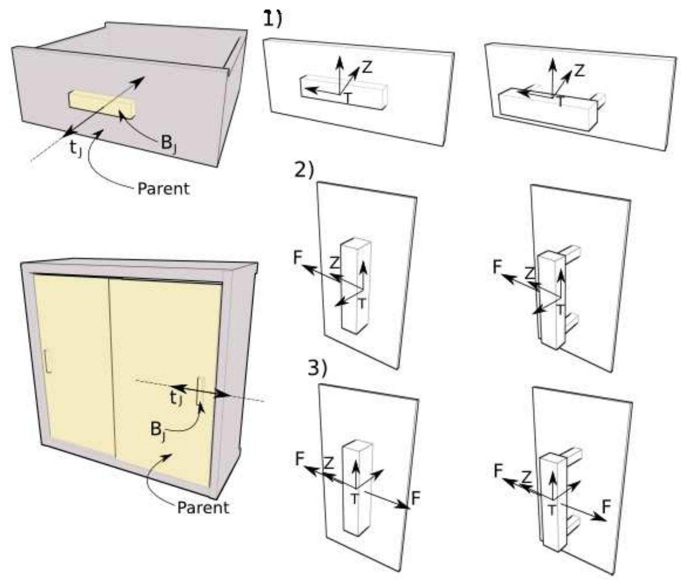

Fig. 4. Task-oriented grasp planning examples.

V. Task planning

The goal of the task planning is to perform the task. This involves two operations: moving the robot from its current position to the grasp position (reaching), and doing the actual task:

A. Reaching the object

For reaching the object, the robot must move the task frame (i.e. its hand) towards the target frame. The difference between both frames is computed as ${}^{\mathcal{TF}}T_{\mathcal{T}} = {}^{\mathcal{TF}}T_{\mathcal{E}} \cdot {}^{\mathcal{E}}T_{\mathcal{R}} \cdot ({}^{\mathcal{W}}T_{\mathcal{R}})^{-1} \cdot {}^{\mathcal{W}}T_{\mathcal{O}} \cdot {}^{\mathcal{O}}T_{\mathcal{T}}$, where $\mathcal{TF}$, $\mathcal{E}$, $\mathcal{R}$, $\mathcal{W}$, $\mathcal{O}$ and $\mathcal{T}$ are the task frame, the end-effector frame, the robot base frame, the world frame, the object frame and the target frame respectively. Of these, the most difficult to obtain are the poses of the object and the robot with respect to the world frame. Localization algorithms for mobile robots can be used for this aim. On the other hand, the target frame with respect to the object main frame, ${}^{\mathcal{O}}T_{\mathcal{T}}$, can be obtained through the object hierarchy, whereas the relations between the robot's base frame, the end-effector frame and the task frame are given by robot's kinematics. In the future, we plan to use vision-based pose estimation algorithms in order to directly obtain ${}^{\mathcal{TF}}T_{\mathcal{T}}$ under unstructured environments.

Once the relation between $\mathcal{TF}$ and $\mathcal{T}$ is known, the simpler strategy for reaching the object is to move the hand along the straight line that links both frames. But in real life, this would result in a collision for certain hand configurations where fingertips must be carefully inserted into the handle. For avoiding this problem, we make use of an approaching direction as other authors have already done [19]. Once more, the task frame is a great help for defining this direction. With the task frames defined as in Figure 2, the approaching direction will always be the Z axis.

Then, reaching is done in two steps: first, the robot moves along a straight line to the point $(0, 0, -100\text{mm})$ in the target frame, and finally the target frame is reached along the approaching direction. In the case of the precision and cylindrical grasps, the fingers need to be opened in suitable way for the given object. In both cases, when the task frame coincides with the target frame, fingers are closed until contact is made. Note that, as the fingers close, the position of the TF with respect to the grasp frame may change. But as the robot continuously tries to align both frames, the hand will move as the fingers close, thus ensuring a good grasp.

B. Performing the task

Once the hand has reached the grasp position, the task must be performed. As explained in Section IV-B, the task is given as a DOF to activate, expressed in a frame of the object's hierarchy, known as the compliance frame $\mathcal{C}$. The relation between $\mathcal{C}$ and $\mathcal{T}$ is known by exploring the object hierarchy. At the grasp position, frames $\mathcal{T}$ and $\mathcal{TF}$ are the same, thus the position of the compliance frame with respect to the robot's hand is known: ${}^{\mathcal{TF}}T_{\mathcal{C}}$. The DOF to activate can be easily transformed into a screw vector on $\mathcal{C}$, and then transformed to $\mathcal{TF}$ through the screw transformation matrix associated to ${}^{\mathcal{TF}}T_{\mathcal{C}}$. In practice, what we do is to unlink the TF from the robot's hand and to set it to the compliance frame.

Modelization errors can occur in practice, and the computed transformations between frames may not correspond exactly to the real values. For this reason, it is very important to perform the task in a compliant manner [20]. The robot is endowed with a wrist force sensor that is used to monitor task forces. High forces (until a value suitable for the Barrett Hand) are allowed opposing the task direction. For directions tangent to the task, a small force makes the robot to correct its trajectory by lateral movements. This will be better appreciated in the example of the next section.

VI. Implementation and manipulation example

Our approach has been implemented using the architecture for compliant execution of manipulation tasks, on the UJI Service Robot [20]. This architecture allows the definition and compliant execution of complex robot manipulation tasks within the Task Frame Formalism (TFF) [8]. It supports the three different ways of specifying the task frame [14] and allows to switch between them at runtime. For safety and performance reasons, the robot is endowed with a compliant impedance velocity/force controller that takes care of external forces, making it robust to environmental changes or modelling errors. Under this architecture, there are perceptions, actions and abilities. The structural model of

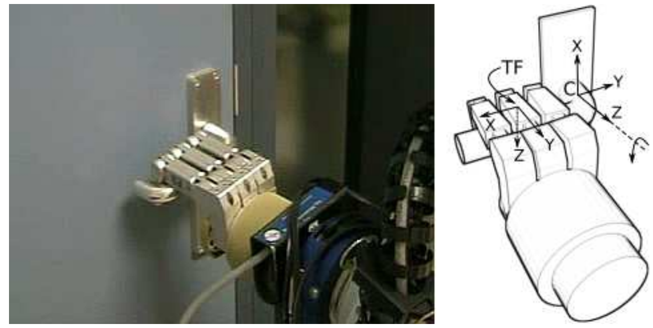

Fig. 5. Frames during the task execution.

the objects, as well as the grasp planning algorithm have been implemented as perceptions. The task planning algorithm is an ability, because it takes as input a set of perceptions (object pose, task, required preshape, etc.), and gives as output an ability network, composed of action primitives and other abilities [20].

We would like to show the validity of our task-oriented grasp planning and task planning algorithms by the execution of a common task such as turning a door handle. Other authors have already addressed this problem (see [12] for an example), but under the task planning perspective. They assume that a grasp has already been found, and focus on the compliant task execution part. As far as we know, there are no integrated solutions that consider the grasp and the task as a whole.

In our experiment, the robot is requested to turn the door handle of a door in our lab. It first retrieves the simplified 3D model of the door from a database (see Figure 3) and finds the frame where the task is defined (frame $\mathcal{H}$ in Figure 3). With this input, it executes the grasp planning algorithm of Section IV-B. With the door example, composed of two boxes and two frames, the execution takes less than 10 milliseconds on a standard Pentium 4 at 3Ghz running GNU/Linux. Following the steps of Section IV-B, the algorithm decides that the hook power preshape is the most suitable for turning the handle, and sets the target frame to the middle of the upper face, with Z direction pointing downwards.

The task planning algorithm selects the reaching strategy. It first moves the robot's hand to a point over the handle, and then reaches the handle through the approaching direction. When contact is detected, the task planning algorithm sets the task frame to the compliance frame (see Figure 5) and the robot is controlled to perform the task by means of the compliant velocity/force controller presented in [20]. Figure 5 shows the robot performing the task.

Figure 6 depicts the forces that appear during the task. First, the absolute value of the force in Z direction increases until an approximate value of 10N which corresponds to the resistance of the particular door handle. Consequently, the hand's velocity along this direction decreases, because the current force is approaching the maximum allowed force for the task direction, which was set to be 15N for the Barrett

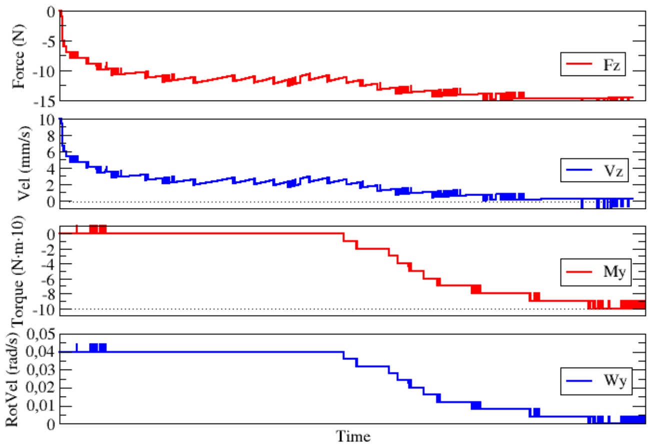

Fig. 6. Forces during the task execution.

Hand. During this stage there is no torque around Y direction which means that the estimation of the compliance frame according to the model is good enough. Then, the handle starts offering more resistance, and the opposite force finally reaches the value of 15N. At this point, as expected, the velocity decreases to zero, and the task is considered as finished. It is worth noting that, during the second stage, the torque around Y axis starts increasing (in negative direction). The robot, then, modifies its velocity around this axis, and stops the motion when the absolute value of the torque reaches the maximum allowed value.

VII. Conclusions and Future Work

Existing grasp planning algorithms do not take the task into account, and current task planning works do not consider the grasp. We think that the task is fundamental when looking for a grasp. An approach for integrating grasp and task planning has been presented. A simplified geometrical and structural description of the object to manipulate, as well as the task to perform are taken as input by a task-oriented grasp planning algorithm which computes the most suitable grasp for the given task. For this, the concept of hand preshape [7] is improved with the inclusion of a task frame [8], and a set of task-oriented hand preshapes for the Barrett Hand and home tasks are defined and classified according to the grasp wrench space they generate

We test our algorithms by making the robot perform a common task such as turning a door handle. First, it has to find a grasp on the door handle suitable for the opening task. Then, the robot plans a path in order to reach the door handle and performs the grasp. Finally, the task is executed taking into account task forces due to modelling errors. As far as we know, this is the first work that considers grasp planning and task planning as a whole, thus establishing the bases for a multipurpose robotic assistant.

As future lines, we would like to test our algorithms with different objects and tasks, under induced modelling errors, in order to study how the system behaves under these errors. We would also like to make the system suitable for unstructured scenarios, with the use of vision for object recognition and pose estimation. Our ultimate aim is to contribute to the evolution of service robots and its final inclusion at homes.

Acknowledgment

The authors would like to thank the Spanish Ministry (MEC), under project DPI2004-01920, and Generalitat Valenciana, under projects CTBPRB/2005/052 and GV05/137 for their invaluable support in this research.

References

[1] A. Bicchi and V. Kumar. Robotic grasping and contact: A review. In *Proc. IEEE Int. Conf. on Robotics and Automation*, pages 348–353, San Francisco, CA, 2000.

[2] C. Ferrari and J. Canny. Planning optimal grasps. In *IEEE International Conference on Robotics and Automation*, pages 2290–2295 vol.3, Nice, France, 1992.

[3] Andrew T. Miller, Steffen Knoop, Henrik I. Christensen, and Peter K. Allen. Automatic grasp planning using shape primitives. In *Proc. IEEE Intl. Conference on Robotics and Automation*, pages 1824–1829, 2003.

[4] Z. Li and S. Sastry. Task oriented optimal grasping by multifingered robot hands. In *ICRA*, pages 389–394 vol.4, 1987.

[5] Ch. Borst, M. Fischer, and G. Hirzinger. Grasp Planning: How to Choose a Suitable Task Wrench Space. In *Proc. IEEE Intl. Conference on Robotics and Automation*, pages 319–325, New Orleans, LA, USA, April 2004.

[6] Robert Haschke, Jochen J. Steil, Ingo Steuwer, and Helge Ritter. Task-oriented quality measures for dextrous grasping. In *Proc. Conference on Computational Intelligence in Robotics and Automation*, Espoo, 2005. IEEE.

[7] D. Wren and R. Fisher. Dextrous hand grasping strategies using preshapes and digit trajectories. In *Proc. IEEE Int. Conf. on Systems, Man and Cybernetics*, 1995.

[8] Matthew Mason. Compliance and force control for computer-controlled manipulators. *IEEE Trans on Systems, Man, and Cybernetics*, 11(6):418–432, 1981.

[9] Barrett Technology Inc. http://www.barrett.com.

[10] N. Pollard. Synthesizing grasps from generalized prototypes. In *Proc. IEEE Intl. Conference on Robotics and Automation*, Minneapolis, Minnesota, April 1996.

[11] J. Napier. The prehensile movements of the human hand. *Journal of Bone and Joint Surgery*, page 38B(4), 1956.

[12] Lars Petersson, David Austin, and Danica Kragic. High-level control of a mobile manipulator for door opening. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 2333–2338, vol 3, Takamatsu, Kagawa, Japan, October 2000.

[13] U. Thomas, B. Finkemeyer, T. Kröger, and F.M. Wahl. Error-tolerant execution of complex robot tasks based on skill primitives. In *Proc. of the 2003 IEEE International Conference on Robotics and Automation*, Taipei, Taiwan, September 2003.

[14] T. Kröger, B. Finkemeyer, U. Thomas, and F.M. Wahl. Compliant motion programming: The task frame formalism revisited. In *Mechatronics & Robotics*, Aachen, Germany, September 2004.

[15] Johan Baeten, Herman Bruyninckx, and Joris De Schutter. Integrated vision/force robotic servoing in the task frame formalism. *International Journal of Robotics Research*, 22(10-11):941–954, 2003.

[16] F. Marrone, F. Raimondi, and M. Strobel. Compliant interaction of a domestic service robot with a human and the environment. In *Proc. of 33rd Int. Symposium on Robotics*, Stockholm, October 2002.

[17] U. Hillenbrand, B. Brunner, Ch. Borst, and G. Hirzinger. The robutler: a vision-controlled hand-arm system for manipulating bottles and glasses. In *35th International Symposium on Robotics*, Paris, France, 2004.

[18] Ch. Borst, M. Fischer, and G. Hirzinger. Grasping the dice by dicing the grasp. In *Proc. of the IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*, Las Vegas, Nevada, USA, October 2003.

[19] Takashi Suehiro Tsutomu Hasegawa and Kunikatsu Takase. A model-based manipulation system with skill-based execution. volume 2, 1992.

[20] M. Prats, A.P. del Pobil, and P.J. Sanz. A control architecture for compliant execution of manipulation tasks. In *Proc. of International Conference on Intelligent Robots and Systems*, Beijing, China, October 2006.