A Sensor-based Approach for Physical Interaction Based on Hand, Grasp and Task Frames

Mario Prats, Pedro J. Sanz and Angel P. del Pobil
Robotic Intelligence Lab
Jaume-I University, Castellón, Spain.
Email: mprats@uji.es

***Abstract*— Robotic manipulation of everyday objects and execution of household chores is one of the most desired (and challenging) skills for future service robots. Most of the current research in robotic grasping is limited to pick-and-place tasks, without paying attention to the whole range of different tasks needed in human environments, such as opening doors, interacting with furniture, household electrical appliances, etc. In this article, a new framework is presented, extending the well established *Task Frame Formalism* (TFF) [1] with new elements that allow to integrate grasp and task into a common approach. The grasp is defined as a desired task-suitable relationship between the robot hand and the object being manipulated. The task is defined under the TFF, which allows to specify tasks for sensor-guided compliant interaction. Some guidelines for sensor-based execution of tasks defined under the proposed framework are also given. Two different examples of manipulation tasks are presented, making use of the proposed approach and disparate sensor information: door opening by vision and force control, and book grasping by tactile and force integration.**

I. INTRODUCTION

Autonomous robots need advanced manipulation skills in order to be useful for the end-user [2]. Most of current research in robotic manipulation is limited to pick and place tasks, without paying attention to the whole range of different tasks needed in human environments. Apart from grasping objects for pick and place, a service robot working in cooperation with humans needs a complete repertoire of tasks, including opening doors, interacting with furniture and household electrical appliances, switching on/off the lights, etc.

Most of the research in robotic grasping community aims at finding a set of contacts on the object in order to obtain force-closure grasps [3]. Force-closure guarantees that the grasp can compensate forces in any direction, but is a too restrictive condition in the sense that it would be much more natural to plan a grasp which can generate the force required for the task, instead of all the possible forces. This is known in the literature as task-oriented grasping, and has received very little attention [4, 5, 6]. However, the grasp depends completely on the intended task, and vice versa. At the same time that the task dictates the way the hand must be arranged around an object, also the grasp dictates the actions that can be safely performed with it.

Our purpose is to develop an approach where grasp and task are jointly considered in a general framework, based on multisensor information for real-time and real-life dependable physical interaction. In this framework, the grasp and the task are represented in terms of hand, grasp and task frames. The grasp is defined as a desired task-suitable relationship between the robot hand and the object being manipulated, whereas the task is defined under the well established *Task Frame Formalism* [1, 7, 8], as a desired motion that must be applied to the object. The concept of *grasp frame*, introduced by [9], along with the concept of *hand frame*, are used for relating the grasp with the task into a common framework. On the one hand, the grasp frame is used as the goal for hand control. On the other hand, it is related to the task, through the object structural model. The grasp frame allows to transform the desired task motion, given in object coordinates, to robot motion, given in robot coordinates, as long as a suitable sensor-based estimation of the hand-to-object relative pose is provided, in order to overcome execution problems due to modelling errors, grasp uncertainties, sliding, etc. Having a good estimation of the hand-to-object pose, the task frame can be estimated in robot coordinates during execution, following a sensor-based task frame tracking approach [1], allowing the robot to adapt its motion to the particular object mechanism, even if no detailed model is present. Two examples of sensor-guided compliant physical interaction tasks, based on the proposed framework, are presented.

Although the concepts of task, grasp and hand frames are not new, they have never been considered into a common approach. To the best of our knowledge there are no practical approaches in the robotics community that consider the grasp and the task as a related problem in a sensor-based control framework. This may be a reason of the few contributions found in task-oriented grasping. The purpose of our approach is to motivate task-oriented grasping by answering the following fundamental questions:

- How can everyday tasks be specified in a common framework, including both the grasp and the task, and allowing for sensor-based control?
- How can a robot plannify a physical interaction task, from the grasping part to task execution, making use of this framework?
- How can a robot combine its sensors and control its motors for performing the grasp and the task in a dependable manner?

In section II, the sensor-based framework for physical interaction is defined. Section III gives some hints for task-oriented grasp planning and sensor-guided task execution. In sections IV and V, a door opening task combining force and visual feedback, and a book grasping task combining force and tactile sensors are presented. Conclusions and future lines are given in section VI.

II. A FRAMEWORK FOR PHYSICAL INTERACTION

Our framework for describing physical interaction tasks is based on the *Task Frame Formalism* (TFF), because of its suitability for all kinds of force-controlled actions. It was first devised by Mason [7], and then reviewed in [1]. In this formalism, the *task frame* is defined as a cartesian coordinate system, given in object coordinates, where the task is defined in terms of velocity and force references, according to the natural constraints imposed by the environment. The task frame is a concept widely used in task planning and control [10, 8]. However, its relation with the grasp has never been considered. In our framework, we extend the task frame with the concepts of *hand* and *grasp* frame, which are used as auxiliary entities for relating the task with the grasp in a common framework. This approach opens the door to a new problem of unified grasp and task planning that will be addressed in the next point, allowing for purposive grasp execution, as well as to perform the task in a grasp-dependent manner.

Regarding grasp planning, research can be classified into two groups: analytical and qualitative approaches. The analytical approach usually makes use of a detailed model of the object and plans a desired contact point and contact force for each of the fingers [11]. The main problem of this approach is the difficulty to perform these grasps in real robotic systems with constrained robotic hands. The qualitative approach defines the grasp as a predefined hand posture (hand preshape) applied to the object along a given approaching direction [12, 13]. This approach is much more suitable for practical implementation on real robots and it is the one adopted in the examples of this work. The concept of *grasp frame* [9] is revisited, and plays a crucial role in this framework: the grasp frame is the bridge between the grasp and the task.

A. Task frame, hand frame and grasp frame

We make use of three different frames for task-oriented grasping: the *task frame*, the *hand frame* and the *grasp frame* (see Figure 1).

The task frame (T) is a frame given in object coordinates, thus linked to the *object frame* (O), where the task is specified according to the TFF [1]. The programmer has to choose a suitable task frame, where the axis match the *natural constraints* imposed by the environment.

The hand frame (H) is a frame attached to the robot hand (or tool) and it is used for control. It is also related with the control strategy used for making contact. As the control is done at the hand frame, it is necessary to link it with the robot *end-effector frame* (E), normally through robot hand kinematics. In the case of a robot holding a tool [2], the hand frame could be placed in the tool tip, but the tool model and pose estimation techniques should be used in order to estimate the hand frame pose with respect to the end-effector. The hand frame can be seen as a particular *feature frame*, as defined in [14]. As stated by the authors, a feature frame can indicate either a *physical entity*, like the fingertip surface for example, or an *abstract geometry property*, as, for example, the middle point between thumb and index finger in opposition.

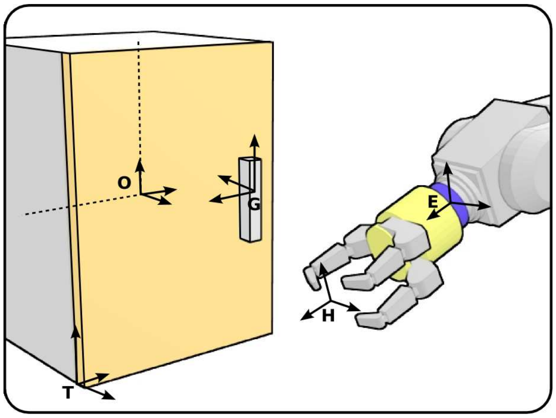


Fig. 1. Considered frames: Task frame (T), grasp frame (G), hand frame (H), object frame (O) and end-effector frame (E)

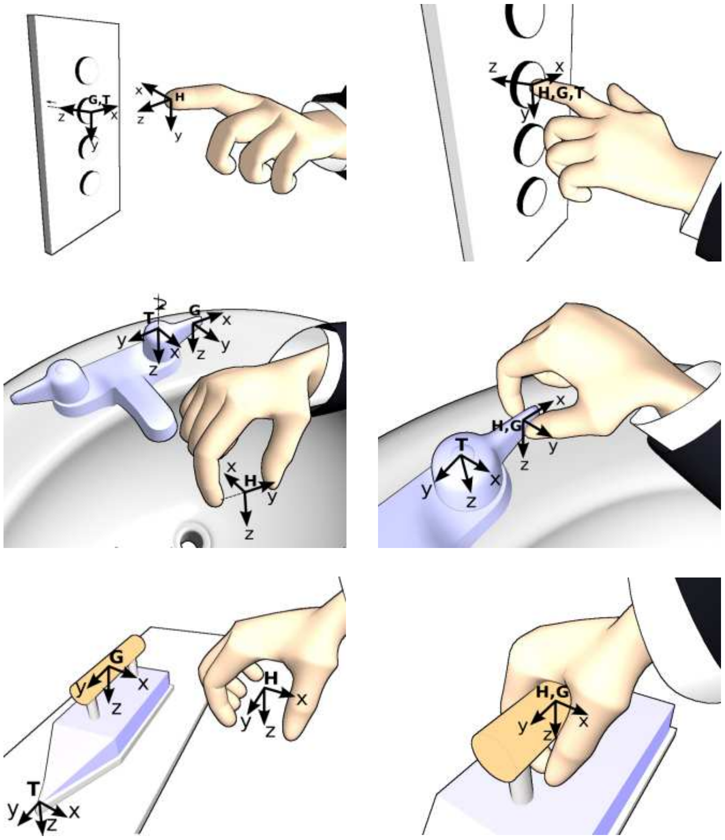
The grasp frame (G) is a frame given in object coordinates, and related to the task frame through object kinematics. This frame is set to parts of the object which are suitable for grasping and task execution. It can also be a *physical entity*, like a button surface, or an *abstract geometry property*, like the symmetry axis of a handle.

The *task-oriented grasp* is then defined as a desired relative pose (possibly under-constrained) between the hand frame and the grasp frame. If this desired relative pose is achieved, the *task*, defined in the task frame, can be transformed to the hand frame, through the grasp frame, allowing the robot to make the motion needed for the task.

B. The framework

In our framework, a task-oriented grasp is any kind of contact between the robot system and the environment, capable of transmitting a force. More concretely, a task-oriented grasp is defined as a desired relative positioning (6 DOFs) between the hand frame and the grasp frame. Constrained and free degrees of freedom for the grasp are also indicated. For the constrained DOFs, the hand frame must completely reach the desired relative pose with respect to the grasp frame. However, for free degrees of freedom, there is no particular relative pose used as reference. Instead, the robot may select a suitable pose, according to manipulability, joint limit avoidance, etc. For example, for pushing a button, a rotation around the normal to the contact surface may be considered as a free DOF.

Let T, G, H and E be the task, grasp, hand and end-effector frames respectively. ${}^E\mathbf{M}_H$, ${}^G\mathbf{M}_T$ and ${}^H\mathbf{M}_G$ are homogeneous matrices relating end-effector frame to hand frame, grasp frame to task frame and hand frame to grasp frame respectively, being ${}^i\mathbf{M}_j = \begin{bmatrix} {}^i\mathbf{R}_j & {}^i\mathbf{t}_j \end{bmatrix}$, where ${}^i\mathbf{R}_j$ is



$\mathcal{P}$ = "*one-finger preshape*"
$^{H}\mathbf{M}_G = \mathbf{I}_{4\times4}$
$\mathbf{S_c} = \text{diag}\,(1,1,1,1,1,0)$
$\mathbf{S_f} = \text{diag}\,(0,0,1,0,0,0)$
$\mathbf{v}^* = (0,0,0,0,0,0)$
$\mathbf{f}^* = (0,0,10\text{ N},0,0,0)$

$\mathcal{P}$ = "*precision preshape*"
$^{H}\mathbf{M}_G = \mathbf{I}_{4\times4}$
$\mathbf{S_c} = \text{diag}\,(1,1,1,0,1,1)$
$\mathbf{S_f} = \text{diag}\,(0,0,0,0,0,0)$
$\mathbf{v}^* = (0,0,0,0,0,0.01\text{ rad/s})$
$\mathbf{f}^* = (0,0,0,0,0,0)$

$\mathcal{P}$ = "*power preshape*"
$^{H}\mathbf{M}_G = \mathbf{I}_{4\times4}$
$\mathbf{S_c} = \text{diag}\,(1,1,1,1,0,1)$
$\mathbf{S_f} = \text{diag}\,(0,0,1,0,0,0)$
$\mathbf{v}^* = \mathbf{f}\,(t)$
$\mathbf{f}^* = (0,0,10\text{ N},0,0,0)$

Fig. 2. Some task examples supported by the task-oriented grasping framework. First: pushing a button, with a force reference. Second: turning on a tap, with a velocity reference. Third: ironing task, with a velocity and force reference.

the 3×3 rotation matrix between frames i and j, and ${}^{i}\mathbf{t}_j$ represents the position of frame j with respect to frame i. Let $\mathcal{P} = \{m_0, m_1, \ldots, m_n\}$ be the hand posture, m_i being the angle for each of the n motors of the hand.

A task-oriented grasp is defined as:

$$\mathcal{G} = \left\{\mathcal{P}, H, G, {}^{H}\mathbf{M}_G, \mathbf{S_c}\right\} \tag{1}$$

where $\mathbf{S_c}$ is a 6×6 diagonal selection matrix which indicates the controlled degrees of freedom for the task-oriented grasp.

The task is defined as a velocity/force reference in the task frame:

$$\mathcal{T} = \{T, \mathbf{v}^*, \mathbf{f}^*, \mathbf{S_f}\} \tag{2}$$

where $\mathbf{S_f}$ is a 6×6 diagonal selection matrix, where a value of 1 at the diagonal element i indicates that the corresponding DOF is controlled with a force reference, whereas a value of 0 indicates it is controlled with a velocity reference. A velocity reference is suitable for tasks where a desired motion is expected, whereas a force reference is preferred for dynamic interaction with the environment, where no object motion is expected, but a force must be applied (for polishing a surface, for example). $\mathbf{v}^*$ and $\mathbf{f}^*$ are, respectively, the velocity and force reference vectors.

A suitable force controller must convert the force references on force-controlled DOFs to velocities, so that the task is finally described as a desired velocity given in the task frame: τ_T^*. For task execution, the desired velocity τ_T^* is converted from the task frame, to the robot end-effector frame as:

$$\tau_E = {}^{E}\mathbf{W}_H \cdot {}^{H}\widehat{\mathbf{W}}_G \cdot {}^{G}\mathbf{W}_T \cdot \tau_T^* \tag{3}$$

where ${}^{i}\mathbf{W}_j$ is the 6×6 screw transformation matrix associated to ${}^{i}\mathbf{M}_j$ [15].

Whereas ${}^{E}\mathbf{M}_H$ and ${}^{G}\mathbf{M}_T$ can be computed from robot kinematics and object model respectively (see Section III), ${}^{H}\widehat{\mathbf{M}}_G$ (the estimated relative pose between the robot hand and the part of the object being manipulated) depends on the particular execution and should be estimated online by the robot sensors. The error between the desired relative pose, ${}^{H}\mathbf{M}_G$, and the estimated pose, ${}^{H}\widehat{\mathbf{M}}_G$, can be due to execution errors such as bad positioning, poor sensory information, sliding, etc. and can be seen as a grasp quality measure. In this sense, the robot must always estimate the grasp quality during task execution in order to constantly improve the grasp, by means of the model, world knowledge, vision sensors, tactile sensors, force feedback, etc. The task frame, according to its definition, must be always aligned with the natural decomposition of the task. Thus, sensors must provide an estimation of the task frame position and orientation during task execution (sensor-based tracking of the task frame [1]). The estimation of $\widehat{{}^{H}\mathbf{M}_{G}}$ is the key for computing the task frame in robot coordinates, thus allowing the transformation of the task specification into robot motion.

Figure 2 shows three examples of daily tasks that can be specified with the proposed framework. The first is an example of a task where a dynamic interaction with the environment is desired. Instead of specifying a velocity, the task is described as a desired force to apply to a button, along Z axis of the task frame T. The hand frame is set to the fingertip, so that it is used to make contact with the button, where the grasp frame, G, has been placed. For this example, the robot may choose the most suitable rotation around Z axis of the hand frame. Thus, this motion is set to be a free DOF.

In the second example, a rotation velocity about Z axis of the task frame, T, is desired in order to turn on the tap. The grasp frame, G, is set to a part suitable for grasping, whereas the hand frame is set to the middle point between thumb and index fingers in a precision preshape. For performing the grasp, the hand frame must match with the grasp frame, up to a rotation about Y axis, which is set to be a free DOF.

Finally, the third example shows a task (ironing) where both a velocity and a force reference is needed. Axis Z of the task frame, T, is force-controlled in order to make some force against the ironing board. At the same time, axis X and Y are velocity-controlled in order to follow a particular trajectory, $\mathbf{f}(t)$. Regarding the grasp, a power preshape is adopted, with a free DOF around Y axis of the hand frame, H.

III. TASK-ORIENTED GRASP PLANNING AND EXECUTION

Usually, it is the programmer who specifies the task in advance according to the requirements. However, for robots designed to work autonomously in home environments, it is desirable to provide an automatic way to build the necessary control entities, such as task frame, grasp frame, force and velocity references, etc. In this section, a task-oriented grasp planning and execution methodology, based on the proposed framework, is presented. Our goal is not to describe here a complete grasp planning algorithm, but to give some guidelines about how to use the proposed framework for the specification and sensor-guided execution of interaction tasks.

A. Task-oriented grasp planning

1) Planning the task frame: For autonomously planning the task, the robot must know the current state of the world, and the state to reach after manipulation. The plan must describe clearly the desired motion that must be applied to the world objects, so that the task frame and force/velocity references are set naturally according to the natural constraints. It can be difficult to find a general method for automatically setting the task frame for all kind of tasks. However, if we consider manipulation of everyday articulated objects with translational and revolute joints, such as doors, drawers, buttons, etc. the task frame can be set naturally from the object *structural model*.

By structural model we mean a set of different object parts that are assembled together. Each part can be defined on its own reference frame, which is independent from the other parts. A set of relations can defined between the parts, in terms of constrained and free degrees of freedom, i.e. a motion constraint can defined with each frame. With this approach, each of the frames defining the structure of the object can be used as the task frame.

As an example, Figure 3 shows a door structural model. It is composed of two parts: the door table, defined in frame O -which is also the object reference frame- and the handle, defined in frame O'. The relation between the handle and the door table can be known, and represented as an homogeneous transformation matrix ${}^{O}\mathbf{M}'_{O}$. The model can also include the degrees of freedom (motion constraint) for each part. In the example of Figure 3, the frame O' is fixed with respect to O, but the frame O has one degree of freedom: a rotation around Y axis, which corresponds to the task of opening the door. Thus, the task can be naturally specified to the robot by means of a frame in the object hierarchy (the task frame) and the degree of freedom that must be activated on it.

2) Planning the hand posture and hand frame: The grasp planning algorithm must ensure that the hand posture is appropriate for generating the desired force on the object through the task-oriented grasp. The hand frame should be set to a part of the hand (or tool) so that the reaching process (moving the hand towards the grasp frame) is done naturally. For example, for pushing a button, the hand frame could be set to the fingertip that would be used for making contact (*physical entity*). However, for a power grasp on a handle, it would be more natural to set the hand frame to the middlepoint between the fingertips and the palm (the grasp centre, an *abstract geometry property*), as shown in Figure 2 (ironing task).

3) Planning the grasp frame: The grasp frame must be set to a part of the object suitable for performing the desired task motion. Normally, the planner should look for handles in the case of big objects, or appropriate contact surfaces for small objects, although the choice of a particular grasp frame depends on the hand preshape and hand frame. The desired relative pose between the hand frame and the grasp frame also depends on the particular choice of both frames, but, normally, it should be set to the identity matrix, as the goal is to align both frames.

B. Task execution

The task execution process can be divided into two stages:

- A reaching/grasping phase, where the hand of the robot must be moved towards the handle until the grasp is executed successfully.
- An interaction phase, where the hand is in contact with the object and the task motion must be performed through robot motion.

The reaching task can be performed by servoing the hand frame towards the grasp frame. It can be done in open loop if a good estimation of the object pose with respect to the robot is available. Closed loop is more adequate if we want to deal with the uncertainties of non-structured environments. Normally, a visual servoing framework is adopted to close the loop during reaching [16].

Regarding the interaction phase, it is worth noting that the robot hand is in contact with the environment, and any kind of uncertainty (errors in the models, bad pose estimation, etc.) may produce very big forces that can damage the environment or the robot. When the robot is in contact with the environment, it is extremely important to design a controller that can deal with unpredicted forces and adapt the hand motion accordingly.

Therefore, a control law based on multiple sensor information, including force feedback, is desired. More concretely, sensors should continuously provide information about the relative pose between the hand (hand frame) and the grasped part (grasp frame). The object or task model can give the relationship between the task and the grasp frame, whereas hand frame pose with respect to the end-effector can be derived from robot hand kinematics. The most important source of error comes from the particular grasp, i.e. from the relationship between the hand and the grasp frame. This relationship must be estimated during execution in order to easily transform the task specification, from object coordinates to robot coordinates.

The best sensor to estimate this relationship is vision. A robot could be observing its hand and the object simultaneously, while applying model-based pose estimation techniques [17]. Another interesting sensor is a tactile array, which provides detailed local information about contact, and could be used to detect grasp mistakes or misalignments. In general, the best solution is to combine several sensor modalities for getting a robust estimation. In the next sections, results on the execution of two different tasks, performed with two different robotic systems under the proposed framework, are presented: one of them (section IV) combines vision and force sensors for opening a door with a parallel jaw gripper, whereas the other (section V) combines tactile and force feedback in order to grasp a book from a bookshelf.

IV. EXPERIMENT I: VISION/FORCE-GUIDED DOOR OPENING

In this section, the task-oriented grasping framework is applied to the task of pulling open the door of a wardrobe, using a mobile manipulator composed of an Amtec 7DOF ultra light weight robot arm mounted on an ActivMedia PowerBot mobile robot. The hand of the robot is a PowerCube parallel jaw gripper. This robot belongs to the Intelligent Systems Research Center (Sungkyunkwan University, South Korea), and is already endowed with recognition and navigation capabilities [18], so that it is able to recognise the object to manipulate and to retrieve its geometrical and structural model from a database.

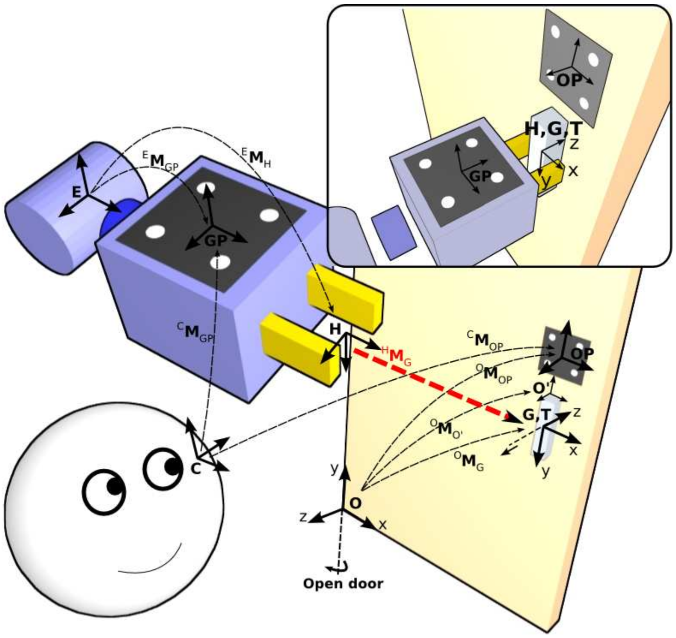


Fig. 3. The vision task is to align hand frame $\mathcal{H}$ and grasp frame $\mathcal{G}$.

A. Planning the task, hand and grasp frame

The structural model of the door is shown in Figure 3. The task of pulling open the door can be specified naturally as a rotation around Y axis of frame O, but also as a negative translation velocity along Z axis of the frame G. The second alternative has the advantage that we can set ${}^{G}\mathbf{M}_{T} = \mathbf{I}_{4\times4}$, without the need to know the door model. We adopt this approach in order to make the solution valid for other doors. Thus, $T = G$, and we set $\mathbf{v}^*$ to be a negative translation velocity along Z axis (the desired opening velocity). As there is no need for force references for this task, $\mathbf{f}^* = \mathbf{0}$ and $\mathbf{S_f} = \mathbf{0}_{6\times6}$.

For the parallel jaw gripper, there are very few manipulation possibilities. We consider only one possible task-oriented hand preshape, which is the precision preshape. The hand frame is set to the middle point between both fingertips, as shown in Figure 3.

As the door contains a handle, the grasp frame is set to the handle, so that the grasp is performed on it. More concretely, the grasp frame is set centered at the handle major axis, as shown in Figure 3. Then, according to the specification of the hand and grasp frames, the desired relationship between both is ${}^{H}\mathbf{M}_{G} = \mathbf{I}_{4\times4}$, i.e. the identity: when grasping, the hand frame must be completely aligned with the grasp frame (the handle must lie in the middle point between both fingertips). For the grasp, a rotation around X axis of the hand frame could be considered as a free DOF. However, as the grip force is very high, we set all the DOFs to be constrained, i.e. $\mathbf{S_c} = \mathbf{I}_{6\times6}$, i.e. the gripper must be always aligned with the handle, as shown in the top right part of Figure 3.

B. *Task execution*

For this task, a position-based visual/force servoing closed-loop approach has been adopted. A robot head observes both the gripper and the object and tries to achieve a relative position between both. This approach has already been adopted in [16], but without considering the subsequent task.

1) Estimating hand-handle relative pose: As already explained in the previous sections, the relationship between the hand and the handle must be estimated continuously during task execution, in order to be able to transform the task motion (given in the task frame) to robot motion (given in the end-effector).

Virtual visual servoing [19] is used to estimate the pose of the hand and the handle, using a set of point features drawn on a pattern whose model and position is known. One pattern is attached to the gripper, in a known position ${}^{E}\mathbf{M}_{GP}$. Another pattern is attached to the object, also in a known position with respect to the object reference frame: ${}^{O}\mathbf{M}_{OP}$. As future research we would like to implement a feature extraction algorithm in order to use natural features of the object instead of the markers. Figure 3 shows the different frames involved in the relative pose estimation process and the task.

The matrix ${}^{H}\widehat{\mathbf{M}}_{G}$, which relates hand and handle, is computed directly from the pose estimation of the gripper and the object, according to the following expression:

$$\left({}^{C}\mathbf{M}_{GP}\cdot{}^{E}\mathbf{M}_{GP}^{-1}\cdot{}^{E}\mathbf{M}_{H}\right)^{-1}\cdot{}^{C}\mathbf{M}_{OP}\cdot{}^{O}\mathbf{M}_{OP}^{-1}\cdot{}^{O}\mathbf{M}_{G} \quad (4)$$

where ${}^{C}\mathbf{M}_{GP}$ is an estimation of the pose of gripper pattern, expressed in the camera frame, and ${}^{C}\mathbf{M}_{OP}$ is an estimation of the object pattern pose, also in the camera frame. ${}^{E}\mathbf{M}_{H}$ and ${}^{O}\mathbf{M}_{G}$ are the hand and grasp frame positions with respect to the end-effector and the object reference frame respectively, as set in the previous points.

2) Improving the grasp: After pose estimation, a measure of the error between the desired (${}^{H}\mathbf{M}_{G}$) and current (${}^{H}\widehat{\mathbf{M}}_{G}$) hand-grasp relative pose is obtained. It is desirable to design a control strategy so that the grasp is continuously improving during task execution. With a vision-based approach, any misalignment between the gripper and the handle (due to sliding, model errors, etc.) can be detected and corrected through a position-based visual servoing control law [20]. We set the vector $\mathbf{s}$ of visual features to be $\mathbf{s} = (\mathbf{t} \quad \mathbf{u}\theta)^T$, where $\mathbf{t}$ is the translational part of the homogeneous matrix ${}^{H}\widehat{\mathbf{M}}_{G}$, and $\mathbf{u}\theta$ is the axis/angle representation of the rotational part of ${}^{H}\widehat{\mathbf{M}}_{G}$. The velocity in the hand frame τ_H is computed using a classical visual servoing control law:

$$\tau_H = -\lambda\mathbf{e} + \widehat{\frac{\partial\mathbf{e}}{\partial t}} \quad (5)$$

where $\mathbf{e}(\mathbf{s},\mathbf{s}^d) = \widehat{\mathbf{L_s}^{+}}(\mathbf{s}-\mathbf{s}^d)$ (in our case, $\mathbf{s}^d = 0$, as ${}^{H}\mathbf{M}_{G} = \mathbf{I}_{4\times4}$). The interaction matrix $\widehat{\mathbf{L_s}}$ is set for the particular case of position-based visual servoing:

$$\widehat{\mathbf{L_s}} = \begin{pmatrix} -\mathbf{I}_{3\times3} & \mathbf{0}_{3\times3} \\ \mathbf{0}_{3\times3} & -\mathbf{L}_w \end{pmatrix}$$

$$\mathbf{L}_w = \mathbf{I}_{3\times3} - \frac{\theta}{2}[\mathbf{u}]_\times + \left(1 - \frac{\mathrm{sinc}(\theta)}{\mathrm{sinc}^2(\frac{\theta}{2})}\right)[\mathbf{u}]_\times^2$$

where $[\mathbf{u}]_\times$ is the skew anti-symmetric matrix for the rotation axis $\mathbf{u}$. Finally, the end-effector motion is computed as $\tau_{\mathbf{E}} = {}^{\mathbf{E}}\mathbf{W}_{\mathbf{H}}\cdot\tau_{\mathbf{H}}$.

3) Task motion and coping with uncertainties: The end-effector velocity that the robot has to achieve in order to perform the task motion, is computed by transforming the task velocity, from the task frame to the end-effector frame, according to equation 3.

Even if the relative pose between the hand and the handle, ${}^{H}\widehat{\mathbf{M}}_{G}$, is estimated and corrected continuously, this estimation can be subject to important errors, considering that it is based on vision algorithms, that can be strongly affected by illumination, camera calibration errors, etc. Due to this fact, the robot motion is also subject to errors, and cannot match exactly the desired motion for the task. As the hand is in contact with the environment, any deviation of the hand motion regarding the task trajectory will generate important forces on the robot hand that must be taken into account.

We adopt an external vision/force control law [21] for integrating vision and force and coping with uncertainties. With this approach, the force vector, with current external forces, is used to create a new vision reference according to:

$$\mathbf{s}^* = \mathbf{s}^d + \widehat{\mathbf{L_s}}\cdot\widehat{\mathbf{L}_\times^{-1}}\cdot\mathbf{K}^{-1}(\mathbf{f}^* - \mathbf{f}) \quad (6)$$

where $\mathbf{f}^*$ is the desired wrench, added as input to the control loop (null in this particular case), $\mathbf{K}$ is the environment stiffness matrix, and $\mathbf{s}^*$ is the modified reference for visual features. $\widehat{\mathbf{L}_\times}$ relates τ_E and $\dot{\mathbf{X}}_E$ according to $\dot{\mathbf{X}}_E = \widehat{\mathbf{L}_\times}\cdot\tau_{\mathcal{E}}$ [20]. Then, the visual servoing control law, described in the previous point, takes as visual reference the new computed reference, $\mathbf{s}^*$.

In conclusion, there are two simultaneous end-effector motions: one, computed by equation 3, which is in charge of performing the task motion, and another one, computed by equation 5, in charge of continuously aligning the hand with the handle by vision/force control. For detailed experimental results of the vision/force-guided door opening task, along with a demonstration video, please refer to [22].

V. EXPERIMENT II: FORCE/TACTILE-GUIDED BOOK GRASPING

Now, the task-oriented grasping framework is applied to the task of taking out a book from a bookshelf, using a mobile manipulator composed of a PA-10 arm, endowed with a three-fingered Barrett Hand, and mounted on an ActivMedia PowerBot mobile robot. The goal of the task is to extract a book from a shelf, while standing among other books. The approach is to do it as humans do: only one of the fingers is used, which is placed on the top corner of the target book

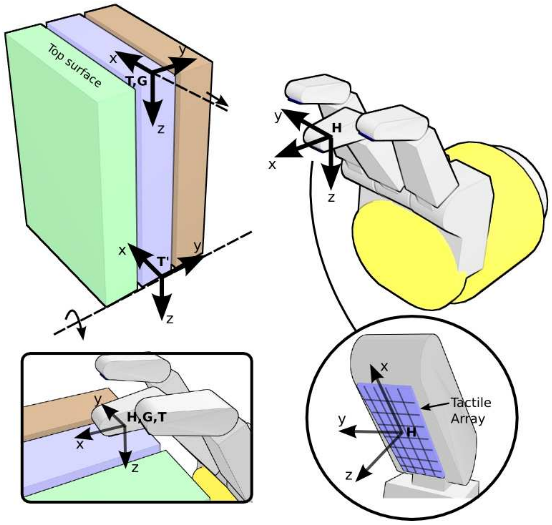


Fig. 4. Frames involved in the book grasping task. The tactile array is used to estimate the relationship between the hand and the grasp frame, ${}^{H}\mathbf{M}_{G}$.

and is used to make contact and pull back the book, making it turn with respect to the base, as shown in Figure 5. In this task, the force/torque sensor is used to apply a force towards the book and avoid sliding, whereas a tactile array provides detailed information about the contact, and helps estimating the hand and grasp frame relationship. As shown in Figure 4, there is one tactile array on each of the fingertips. This sensor consists of an array of 8×5 cells, each of one can measure the local pressure at that point.

A. Planning the task, hand and grasp frame

In Figure 4, a representation of the book grasping task, including the necessary frames, is shown. There are two possibilities for the task frame in this case. The first is to set it to the book base (frame T' in Figure 4), so that the task could be described as a rotation velocity around this frame. The second possibility is to set the task frame on the top edge of the book (frame T in Figure 4), so that the task is described as a negative translational velocity along X direction. We have opted for the second solution, because, in this case, the task frame coincides with the grasp frame, and, then, there is no need to know the book model. In the first case, the height of the book should be known in order to transform the task from the task frame to the hand frame. By adopting the second solution, we make the approach general for any book size. Two references are set in the task frame, $\mathbf{v}^*$ and $\mathbf{f}^*$. The first one is set to a negative velocity in X axis, in order to perform the task motion, whereas $\mathbf{f}^*$ is set to a force along Z axis. This force is needed in order to make enough pressure on the book surface and avoid slip. We have set it to 10 N for our particular system, but it depends on the friction coefficient between the fingertip and the book. For small friction, a bigger force would be needed. Therefore, $\mathbf{S_f}$ is set to $\mathbf{diag}(0, 0, 1, 0, 0, 0)$.

For this task, we define a special hand posture where one of the fingers is slightly more closed than the other ones, so that we can easily make contact on the top of the book with one finger, as shown in Figure 4. The hand frame is set to the inner part of the middle finger fingertip, just in the centre of the tactile sensor. The hand frame pose with respect to the robot end-effector, ${}^{E}\mathbf{M}_{H}$, is computed from hand kinematics.

Fig. 5. The robot grasping the book by means of sensor-based continuous estimation of hand-to-object relative pose.

The fingertip has to make contact on the top of the book. Therefore, we set the grasp frame to the book top surface, which could be located by vision or range sensors. The desired relationship between the hand and the grasp frame, ${}^{H}\mathbf{M}_{G}$, is set to the identity. Although some free DOFs could be set for this contact (rotation in Y to some extent, or even rotation in Z), it is desirable to keep the contact surface as wide as possible in order to increase friction. It is for this reason that all the grasp DOFs have been constrained ($\mathbf{S_c} = \mathbf{I}_{6\times 6}$), so that the fingertip surface is always parallel to the book top surface, ensuring a stable surface contact.

B. Task execution

In this case, the task is performed by combining force and tactile feedback. Tactile information is used to estimate and improve the contact between the hand and the book, whereas force feedback is used in order to cope with uncertainties and ensure that a suitable force is performed on the book surface so that there is no sliding.

1) Estimating hand-book relative pose: Contact on the book is performed with the tactile array. Depending on the sensor cells that are activated, the relative pose between the sensor surface and the book can be estimated. It is not possible to compute the complete relative pose only with tactile sensors, because they only provide local information when there is contact. However, we can obtain a qualitative description of the relative pose. For example, if there is contact with the upper part of the sensor, but not with the lower part, we can deduce that the sensor plane is rotated around Y axis with respect to the book top plane.

All the tactile cells lie in the XY plane of the hand frame. We consider that the finger is completely aligned with the book surface when there are cells activated on each of the four XY quadrants of the hand frame, i.e., all the tactile sensor surface is in contact. If there is contact on the upper half of the sensor, but not on the lower half, or vice versa, we consider that there is a rotation about Y axis, between the sensor (hand frame) and the book surface (grasp frame). Similarly, a rotation around X axis can be detected.

2) Improving the grasp: The goal of this process is to align the finger (tactile sensor) surface with the book surface, taking as input the qualitative description of the relative pose, described in the previous point. We follow a reactive approach, where fingertip rotation around X and Y axis of the hand frame is continuously controlled, in order to obtain contact on each of the XY quadrants of the hand frame. With this approach, the behaviour of the robot is completely reactive to the tactile sensor readings. The goal is to keep the sensor plane always parallel to the book top plane, thus ensuring that ${}^{H}\widehat{\mathbf{M}}_{G} = \mathbf{I}_{4\times 4}$.

3) Task motion and coping with uncertainties: According to the task description, the task motion is performed by moving the hand along negative X axis of the task frame, while applying a force along Z axis. This motion makes the book turn with respect to the base, as shown in Figure 5. Note that, as the fingertip moves backwards and the book turns, the tactile sensor may lose contact with the lower part. This situation is detected by the qualitative pose estimator, and corrected with the control strategy described in the previous point, so that the hand frame is always aligned with the grasp frame, ensuring that task motion can successfully be transformed to end-effector coordinates by equation 3. Figure 5 shows a sequence of the robot performing the task.

VI. CONCLUSION

A new framework for specifying simultaneously the grasp and the task has been proposed, based on the concepts of hand, grasp and task frames. The grasp frame has been introduced in order to translate the task description, given in object coordinates, to the required robot motion. For this, a sensor-based estimation of the relative pose between the robot hand and the object must be continuously available during task execution. Knowing the hand-to-object relationship during execution, the robot can perform the task even with a poor task description or in the presence of inaccuracies, inherent to real life experimentation. Two examples of sensor-guided compliant physical interaction tasks, based on the proposed framework, have been presented: a door opening task by means of vision and force feedback, and a book grasping task which integrates force and tactile information.

As future research, we would like to use the proposed framework for the specification and compliant execution of several common tasks in home environments, based on visual, tactile and force feedback. We think that the integration of multiple and disparate sensor information for hand-to-object pose estimation is a key point for successful and robust robotic physical interaction.

ACKNOWLEDGMENT

The authors would like to thank Universitat Jaume I - Bancaja, under project PI 1B2005-28, and Generalitat Valenciana, under projects CTBPRB/2005/052 and GV-2007-109 for their invaluable support in this research. Authors are very grateful to Prof. Philippe Martinet and Prof. Sukhan Lee for their contribution to the vision-force experiments.

REFERENCES

[1] H. Bruyninckx and J. De Schutter. Specification of force-controlled actions in the 'task frame formalism': A synthesis. *IEEE Trans. on Robotics and Automation*, 12(5):581–589, 1996.
[2] C.C. Kemp, A. Edsinger, and E. Torres-Jara. Challenges for robot manipulation in human environments. *IEEE Robotics & Automation Magazine*, 14:20–29, 2007.
[3] A. Bicchi and V. Kumar. Robotic grasping and contact: A review. In *IEEE Int. Conf. on Robotics and Automation*, pages 348–353, San Francisco, USA, 2000.
[4] Z. Li and S. Sastry. Task oriented optimal grasping by multifingered robot hands. In *IEEE Int. Conf. on Robotics and Automation*, 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 *IEEE Int. Conf. on Robotics and Automation*, pages 319–325, New Orleans, USA, 2004.
[6] R. Haschke, J.J. Steil, I. Steuwer, and H. Ritter. Task-oriented quality measures for dextrous grasping. In *IEEE Conf. on Computational Intelligence in Robotics and Automation*, Espoo, Finland, 2005.
[7] M. Mason. Compliance and force control for computer-controlled manipulators. *IEEE Trans. on Systems, Man, and Cybernetics*, 11(6):418–432, 1981.
[8] J. Baeten, H. Bruyninckx, and J. De Schutter. Integrated vision/force robotic servoing in the task frame formalism. *Int. Journal of Robotics Research*, 22(10-11):941–954, 2003.
[9] T. Hasegawa, T. Suehiro, and K. Takase. A model-based manipulation system with skill-based execution. *IEEE Trans. on Robotics and Automation*, 8(5):535–544, 1992.
[10] M. Stilman. Task constrained motion planning in robot joint space. In *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, pages 3074–3081, 2007.
[11] X. Zhu and J. Wang. Synthesis of force-closure grasps on 3-d objects based on the q distance. *IEEE Trans. Robotics and Automation*, 19(4):669–679, 2003.
[12] A. Morales et al. Integrated grasp planning and visual object localization for a humanoid robot with five-fingered hands. In *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, pages 5663–5668, Beijing, China, 2006.
[13] A.T. Miller, S. Knoop, H.I. Christensen, and P.K. Allen. Automatic grasp planning using shape primitives. In *IEEE Int. Conf. on Robotics and Automation*, pages 1824–1829, Taipei, Taiwan, 2003.
[14] J. De Schutter et al. Constraint-based task specification and estimation for sensor-based robot systems in the presence of geometric uncertainty. *The Int. Journal of Robotics Research*, 26(5):433–455, 2007.
[15] W. Khalil and E. Dombre. *Modeling identification and control of robots*. Hermes Penton Science, 2002.
[16] R.P. Horaud, F. Dornaika, and B. Espiau. Visually guided object grasping. *IEEE Trans. on Robotics and Automation*, 14(4):525–532, 1998.
[17] T. Drummond and R. Cipolla. Real-time visual tracking of complex structures. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(7):932–946, 2002.
[18] S. Lee et al. Robust recognition and pose estimation of 3d objects based on evidence fusion in a sequence of images. In *IEEE Int. Conf. on Robotics and Automation*, Rome, Italy, 2007.
[19] E. Marchand and F. Chaumette. Virtual visual servoing: a framework for real-time augmented reality. In *EUROGRAPHICS 2002*, volume 21(3), pages 289–298, Saarebrücken, Germany, 2002.
[20] P. Martinet and J. Gallice. Position based visual servoing using a nonlinear approach. In *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, volume 1, pages 531–536, Kyongju, Korea, 1999.
[21] Y. Mezouar, M. Prats, and P. Martinet. External hybrid vision/force control. In *Intl. Conference on Advanced Robotics (ICAR'07)*, Jeju, Korea, 2007.
[22] M. Prats, P. Martinet, A.P. del Pobil, and S. Lee. Vision/force control in task-oriented grasping and manipulation. In *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, San Diego, USA, 2007. Video available in proceedings or at: `http://www.robot.uji.es/lab/plone/Members/mprats/video/edited/iros07-video.wmv`.