A FRAMEWORK FOR GRASP SIMULATION AND CONTROL IN DOMESTIC ENVIRONMENTS

Johan Tegin & Jan Wikander

Dept. of Machine Design
Royal Institute of Technology
Stockholm, Sweden

Abstract: A service robot for a domestic environment should be able to grasp, move, and release objects. To successfully grasp objects that cannot be expected to be perfectly modelled and whose positions are not exactly known, control algorithms need to handle uncertainty. Such controllers often use information from force/torque, tactile and joint angle sensors. This paper presents a simulation environment and proposes a design approach for controlling robotic grasping under those conditions. Finally, results from simulation of a sample task are presented.

Keywords: Robotic Manipulators, Control, Simulation, Models, Sensors, Design, Verification, Uncertainty

1. INTRODUCTION

Secure grasping and manipulation of objects in a domestic setting is a necessary skill for a truly useful service robot. In a more or less uncontrolled environment, such as that of a house or apartment, object models cannot be expected to be perfect. In addition, the environment can be dynamic, fragile, and few tasks are repetitive. Designing hardware and control for such conditions is difficult even for "simple" tasks; pick up the remote, open the door, switch the light on, or fetch the cereal box.

In the not too distant future, advances in humanoid development and research will be combined with semi or fully autonomous manipulation capabilities. This will open up new opportunities: Robots that can provide 24-hour household help for the growing group of elders, can perform deep space missions, or assist in hazardous tasks such as mine clearance. Agriculture and food processing are other applications where autonomous grasping can be of great use (Lee, 2000). Safety, reliability, and functional requirements for industrial and domestic robots are very high. For toys it's a different story, perhaps there is were we will first see autonomous grasping in a large scale.

Previously presented results with respect to grasping and manipulation include catching a ball, playing the piano, and more using the robotic DLR Hand (Borst *et al.*, 2003). Exchanging a light bulb has been shown using the Utah/MIT hand (Jägersand, 1997). High speed grasping has also been performed (Namiki *et al.*, 2003). These impressive results and many more, have typically been reached using advanced, specialized hardware and under rather well defined conditions. Performing such advanced tasks in an uncertain environment using less complex hard- and software is even more difficult, if not impossible.

Simpler, fetch-and-carry type tasks can however be performed using more reasonably priced hardware, even in uncertain domestic environments. This paper presents a framework for design of controllers and for simulation of such tasks.

The paper is organized as follows: An example task is presented in the next section. Section 3 presents the overall control approach, the control method and the sensors used. The simulation environment is outlined

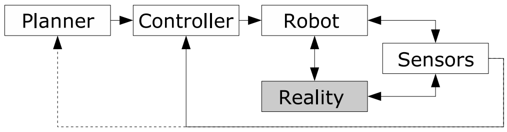

Fig. 1. Physical outline of a grasping system.

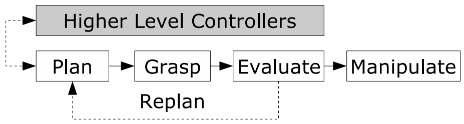

Fig. 2. Algorithm outline for a grasping task.

in Section 4. Results from simulation are presented in Section 5 and eventually Section 6 concludes the paper.

2. THE PROBLEM

Simple manipulation tasks can be executed by securing the object using the hand and perform the actual manipulation moving the robot arm and base. This approach is referred to as grasping and/or fixturing and not to be confused with *dexterous* manipulation where the object motion is a result of finger motions. On a high level of abstraction, most grasping systems follow a physical system layout as shown in Fig. 1 and have a control layout as depicted in Fig. 2.

It is assumed that the object model as well as object position and orientation are not perfectly known. Even if there in addition to models is for example computer vision to aid in refining models and to use in feedback control, objects may for example be occluded by the hand itself. A grasp control algorithm must always be able to handle at least some uncertainty. The proposed controller can handle an error in the object size or position in the centimeter range using *low level control only*. In doing so, there are also certain limitations; the controller does not take advantage of the rest of the system, it cannot reposition the hand, or detect object motion from vision. But the controller becomes highly modular without any need for communication with higher level controllers. This can be seen as creating local autonomy which facilitates system integration.

2.1 Example Task

A typical task in a domestic environment is grasping and lifting an object. This paper shows how the suggested framework can be used to simulate secure lifting of a box. The box is $170 \times 170 \times 60$ mm in size, made from plastic and initially resting on a wood surface. The box is grasped from the side using a Barrett hand (Townsend, 2000) and a three-fingered opposing grasp. Eventually the object is lifted using the Puma arm to which the hand is mounted, see Fig. 3.

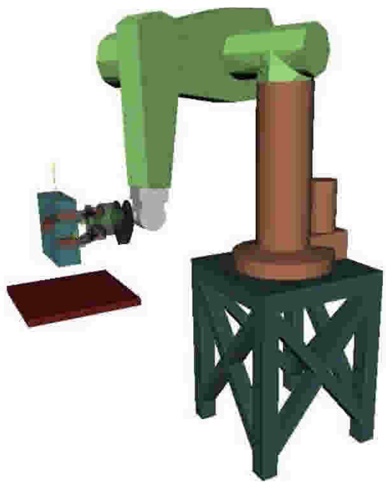

Fig. 3. The box is grasped and lifted by a Barrett hand mounted to a Puma arm.

It is assumed that the desired grasp has already been planned. The problem is thus reduced to control the grasp formation to reach a secure grasp. To simulate some uncertainty the box is placed 20 mm to the left and 10 mm too far away [1] from the "ideal" position. The "ideal" position is where the box is right in front of the hand at a distance such that the fingers contact the vertical centerlines of the box sides.

2.2 Simulation

In the development of grasp controllers it is helpful to be able to simulate the grasping process. There is no need to consider safety aspects or care about hardware reliability. Also, implementing changes to models and controllers is easier. But simulations never consider all real robot issues such as real time properties, control interfaces, noise, hysteresis, drift, et cetera. While a limitation, that is also the advantage using simulations: these factors can be avoided in the early stages of development and later on it can be investigated how these factors affect overall control.

Despite the advantages in using simulation, there is not much software available for grasp simulation. Thus, to allow testing of controllers, a simulation environment was created by expanding and interfacing to the only grasp simulator of our knowledge, GraspIt! (Miller, 2001).

If a high quality dynamic physical modeling is essential, other tools may be more suitable, see e.g. (Ferretti *et al.*, 2006). But since grasping can often be performed rather slowly, and considering that model errors for mass properties, sensors, friction, and in actuator and gear models are often quite large, second order dynamic effects can be ignored in the control design and instead considered as small disturbances. (Prattichizzo and Bicchi, 1998)

[1] View from the palm towards the box.