

# Humanoid Mobile Manipulation Using Controller Refinement

Robert Platt, Robert Burridge, Myron Diftler, Jodi Graf, Mike Goza, Eric Huber

Dexterous Robotics Laboratory

Johnson Space Center, NASA

{robert.platt-1, robert.r.burridge, myron.a.diftler, jodi.s.graf, s.m.goza}@nasa.gov, eric@roboteyes.com

**Abstract**—An important class of mobile manipulation problems are “move-to-grasp” problems where a mobile robot must navigate to and pick up an object. One of the distinguishing features of this class of tasks is its coarse-to-fine structure. Near the beginning of the task, the robot can only sense the target object coarsely or indirectly and make gross motion toward the object. However, after the robot has located and approached the object, the robot must finely control its grasping contacts using precise visual and haptic feedback. In this paper, it is proposed that move-to-grasp problems are naturally solved by a sequence of controllers that iteratively *refines* what ultimately becomes the final solution. This paper introduces the notion of a refining sequence of controllers and characterizes this type of solution. The approach is demonstrated in a move-to-grasp task where Robonaut, the NASA/JSC dexterous humanoid, is mounted on a mobile base and navigates to and picks up a geological sample box. In a series of experiments, it is shown that a refining sequence of controllers decreases variance in robot configuration relative to the sample box until a successful grasp has been achieved.

## I. INTRODUCTION

It is expected that one of the most common tasks for future mobile humanoid robots will be to locate, pick up, and retrieve objects. Indeed, NASA foresees this as one important way that humanoids will be able to assist astronauts on future lunar and planetary missions. Instead of addressing mobile manipulation in general, this paper specifically focuses on *move-to-grasp* problems where a mobile manipulator must locate, approach, and lift a desired object. In addition, it is proposed that move-to-grasp problems are best solved by a *refining* sequence of controllers, where each controller in the sequence iteratively confines the robot to a smaller and smaller region of configuration space.

In the literature, mobile manipulation is frequently equated with solving force and/or motion control tasks with one or more mobile manipulators. Important previous work includes work out of Khatib’s lab regarding the augmented object model for controlling object dynamics in operational space and the virtual linkage model for modeling internal forces [1]. Tan *et al.* demonstrated an approach to kinematic optimization and hybrid position and force control in the context of a cart pushing task using a mobile manipulator attached to a non-holonomic mobile base [2]. MacKenzie and Arkin adapted a behavior-based approach to a drum sampling task where a mobile robot must locate and approach a barrel and insert a probe into its bung hole [3]. Petersson and Christensen divided

the mobile manipulation problem into a mobility portion and a manipulation portion [4]. They proposed that the mobility part is best solved using behavior-based approaches while the manipulation part should be solved using a hybrid dynamical system. Pimentel *et al.* proposed a behavior-based architecture that can be applied to a cooperative carrying task [5].

This paper focuses on sequential control in move-to-grasp problems. Move-to-grasp problems are an interesting subset of mobile manipulation problems because they require the robot to move in a precise way to a small set of configurations. In principle, these problems can be solved by a path planning process that identifies the desired configuration and then moves the robot there. However, in practice, this is difficult because sensor noise impedes precise geometric modelling and actuation error makes it difficult to execute geometric motion plans precisely. A different approach executes a sequence of robust closed-loop controllers to achieve the goal. This approach can be more robust than geometric path planning because, compared with end-to-end planning solutions, it is easier to characterize controllers with limited functionality in open environments. In addition, closed-loop controllers allow the system to use different types of feedback and actuation at different points in the process. This paper notes the existence of a class of problems that can be solved by a *refining sequence* of controllers. A refining sequence satisfies two conditions: 1) each controller in the sequence converges to a configuration within the domain of attraction of the next controller; 2) the domain of attraction of each controller is a subset of the domains of all previous controllers. A refining sequence of controllers iteratively confines the robot to a smaller and smaller region of configuration space. This paper identifies move-to-grasp problems can be solved by a refining control sequence and tests the approach on a move-to-grasp task involving Robonaut, the NASA/JSC dexterous humanoid.

## II. CONTROLLER REFINEMENT

Controller refinement is a special case of controller funneling [6]. In controller funneling, pairs of controllers that execute sequentially must satisfy the *prepares* condition.  $\pi_1$  is said to *prepare*  $\pi_2$  when the goal region of  $\pi_1$  is inside the domain of  $\pi_2$ :  $g(\pi_1) \subseteq \mathcal{D}(\pi_2)$ . This condition guarantees that the robot always remains within the domain of attraction of the currently executing controller. Effectively, these controllers “funnel” the state of the robot toward a goal configuration. A

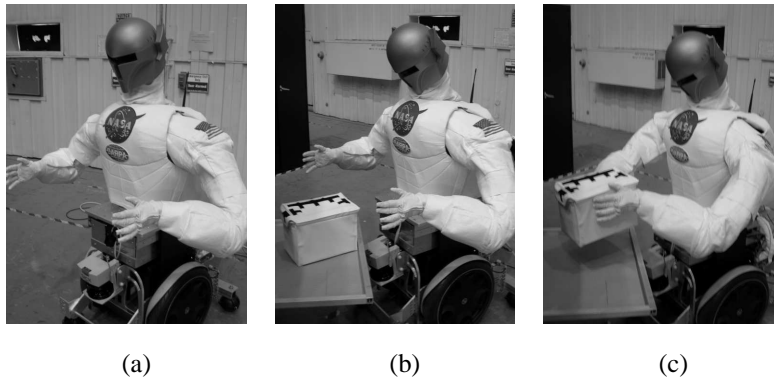


Fig. 1. Illustration of Robonaut completing the move-to-grasp task in the Robonaut-SCOUT field study.

major advantage of this approach is that it is unnecessary to design a single, monolithic controller that converges to the task goal and yet has a large enough domain of attraction. Burridge, Rizzi, and Koditschek demonstrate that controller funneling can be an effective approach to dynamic robot juggling tasks [6]. Controller funneling has also been used in grasp synthesis where two grasp controllers execute sequentially to generate an enveloping grasp [7]. In addition, funneling control sequences that encode quadrupedal walking behavior have been autonomously learned using Reinforcement Learning [8].

Controller refinement defines another constraint in addition to the *prepares* condition. If  $\pi_2$  *refines*  $\pi_1$ , then the domain of attraction of  $\pi_2$  must be a subset of the domain of  $\pi_1$ :  $\mathcal{D}(\pi_2) \subseteq \mathcal{D}(\pi_1)$ . Refining sequences are particularly robust because at any given point during execution of the sequence, the robot is within the domain of attraction of every controller that has executed up to that point. Even if external perturbations push the robot outside of the domain of attraction of the currently executing controller, the robot may “land” within the domain of an earlier controller in the sequence.

Also, when the refining sequence is specified by defining a policy over a discrete state space, a simple state representation exists. Consider the domains of attraction for every controller in the sequence. The pattern of membership of the current robot configuration in the domains of attraction contains sufficient information to decide which controller to execute next. If only the controllers in a single refining sequence are included, then it is sufficient to know only the identity of the smallest domain of attraction that contains the robot’s current configuration. This state representation is similar to that of the control basis [8]. However, instead of representing the pattern of controllers that have already executed, this representation encodes the set of controllers that can execute.

### III. CONTROLLERS

This paper explores controller refinement using the following sequence of six controllers: approach region, approach object, reach toward object, guarded move, comply with object, and lift object. The approach region controller,  $\pi_{ar}$ , navigates over uneven terrain while avoiding obstacles to within 2.5m of

the object to be picked up. A high level controller iteratively computes obstacle-free paths to the goal at approximately 10Hz. The low level controller follows this path by referencing PD controllers to via points along the last computed path. It is assumed that the goal region can be identified by looking for a large object known to be in the vicinity of the target object. In the implementation that this paper reports on, the sample box is assumed to be located on SCOUT. Before moving, Robonaut visually localizes SCOUT, identifies local obstacles using a laser range finder, and plans an obstacle-free path to SCOUT. Robonaut moves by appropriately parameterizing PD controllers that servo to positions and angles along the path. *En route* to SCOUT, approach region updates the positions of local obstacles using the laser range finder and re-evaluates a new obstacle-free path at approximately 10Hz.

After navigation to within 2.5m of the target object, the approach object control policy,  $\pi_{ao}$ , drives toward the object in three stages. When Robonaut is more than 1.8m away from the target object, it drives directly toward the object to a point 1.5m away. Once Robonaut is less than 1.8m away, it drives to a point 1.5m directly in front of the object. Finally, Robonaut drives to a point directly in front of the object. After approaching the object, Robonaut reaches both palms to visually determined reference configurations around the box by executing  $\pi_{reach}$ . This controller reaches the centers of both palms to pre-specified positions and orientations relative to the box.

Next, a guarded move controller,  $\pi_{gm}$ , executes that places both palms in contact with the object. This controller concurrently executes two control primitives, a position controller and a force controller. Executed alone, the position controller moves the centers of both palms to the visually-located center of the box. The force controller complies to applied forces so as to achieve a zero force reference. These two control primitives execute concurrently by projecting the output of the position controller into the null space of the force controller [9]. In the terminology of the control basis, the position controller executes “subject to” the force controller [10]. When no forces are applied to the palm, this controller moves the palms toward the object. However, the controller will not

Step	Controller	Description
1	$\pi_{ao}$	approach object
2	$\pi_{reach}$	reach toward object
3	$\pi_{gm}$	guarded move
4	$\pi_{comply}$	comply to object
5	$\pi_{lift}$	lift object

TABLE I

THE REFINING CONTROL POLICY USED IN THE ROBONAUT-SCOUT FIELD STUDY.

push into the object because the higher-priority force control primitive will prevent the manipulator from applying large forces to the object.

After making contact with the sides of the box, a compliance controller,  $\pi_{comply}$ , flattens Robonaut’s palms against the sides of the box. This controller concurrently executes a moment controller, a force controller, and a position controller using null space controller composition. The position controller has the highest priority and keeps the palm in approximately the same position on the object surface. Without violating this position constraint, the force controller applies an inward force along the object surface normal. Finally, the moment controller executes with the lowest priority, complying a flat palm to the surface of the object. Essentially, the force controller pushes the palm flat against the object. Last, after making good contact with the object, a lift controller,  $\pi_{lift}$ , moves the two palms to a reference position while applying an inward holding force. The highest priority control primitive applies an internal force between the two palms while the subordinate control primitive moves the two palms to the goal.

#### IV. EXPERIMENTS

Controller refinement was explored in the context of a field study involving the NASA/JSC Robonaut mounted on an RMP mobile base and a semi-autonomous rover, SCOUT. Starting far away from SCOUT, Robonaut navigated to a platform mounted on the rear of SCOUT. After reaching the platform, Robonaut picked up a geological sample box.

This task was accomplished by executing the sequence of controllers illustrated in Table I. Because this paper’s experiments were conducted indoors, the approach region controller was omitted and the approach object controller executed first. The approach object control policy moved the RMP directly in front of the object. When the RMP is less than 0.7m from the box, the policy executes a reach controller that moves the hands around the box. Next, the policy executes a guarded move controller that makes contact with the sides of the box. After making contact, the control policy executes a compliance controller that presses the palms against the sides of the box. Finally, a lift controller executes to lift the box.

In order to characterize this solution to the move-to-grasp task, a series of eight experiments were conducted where Robonaut navigated to and picked up a geological sample box measuring 7in  $\times$  8in  $\times$  11in. This is illustrated in Figure 1. In Figure 1(a), Robonaut is 2.25m away from the box. In

Figure 1(b), Robonaut has navigated to a point just in front of the box. In Figure 1(c), Robonaut is lifting the box.

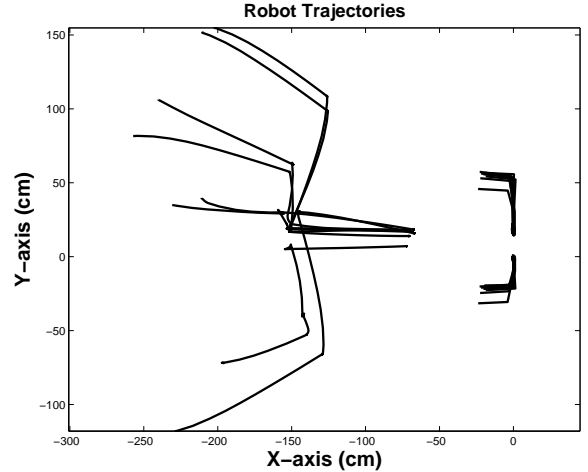


Fig. 2. The trajectories taken by Robonaut during the eight experimental trials. The “lightning-bolt” trajectories on the left side are the trajectories taken by the mobile base. The “L”-shaped trajectories on the right are the paths taken by Robonaut’s two palms.

Figure 2 illustrates the trajectories followed by the robot during these eight trials. In this figure, the sample box is at the origin with its major axis oriented horizontally. The lines on the left side of the plot illustrate the path of the center of the Robonaut RMP base. The two clusters of “L”-shaped lines on the right illustrate the paths of the left and right palms. The “lightning bolt” shape of the RMP trajectories is the result of the approach object control policy. Since, in each of these trials, Robonaut started less than 2.5m from the sample box, Robonaut executes the approach object control policy first and moves directly toward the box. When it gets to a point within 1.5m, Robonaut moves to a point along the axis of the box. When Robonaut reaches a point 1.5m directly in front of the box, the system drives toward the box. After arriving in front of the box, approach object terminates and Robonaut reaches the two palms toward the box. Following the reach, the palms make contact with the sides of box, comply with the box, and pick it up.

The eight trajectories shown in Figure 2 illustrate how Robonaut is confined to a smaller and smaller region of configuration space as it approaches the goal. Robonaut starts the experiment in a large range of positions, approximately 2.25m away from the object. However, the variance in Robonaut’s position decreases significantly when it reaches a position directly in front of the sample box. Finally, after Robonaut makes contact and complies with the box, this variance virtually disappears.

Robonaut’s progression through the refining sequence of controllers is mirrored by a continual decrease in the variance of the estimated pose of the sample box. This is illustrated in Figure 3. When Robonaut is 2.25m away from the box, the variance in the visually estimated position is large (the “approach region” bar in Figure 3). However, after approach-

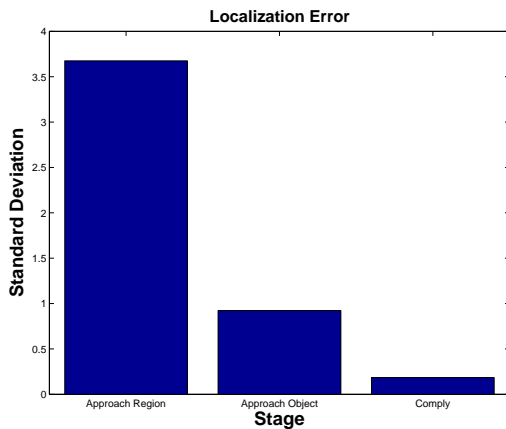


Fig. 3. Standard deviation in the estimated box position decreases as the refining control policy of Table I executes. The first bar, “approach region” gives standard deviation when Robonaut is approximately 2.25m away from the sample box. The second bar shows standard deviation after approaching the sample box. The third bar shows standard deviation after making contact and complying to the sides of the box.

ing the box, Robonaut is able to localize the box much more precisely (the “approach object” bar). Finally, after contacting and complying with the object, Robonaut augments the visual information with tactile information that enables the object pose to be estimated very precisely (“comply” bar).

## V. CONCLUSION

This paper has addressed a class of mobile manipulation problems called “move-to-grasp” problems, where a mobile manipulator must navigate to and pick up an object. It is proposed that move-to-grasp problems are best solved by a *refining* sequence of controllers, where each controller in the sequence iteratively confines the robot to a smaller and smaller region of configuration space. Refining sequences are particularly robust because the robot is always within the domain of attraction of all previously executed controllers in the sequence. This approach is explored in a move-to-grasp task where Robonaut, a dexterous humanoid robot, navigates to and picks up a geological sample box off of a platform in the rear of SCOUT, a semi-autonomous lunar rover. Results are given that show that over a series of experimental trials, the refining sequence of controllers iteratively confines Robonaut’s configuration to a smaller and smaller region around the sample box. On each trial, the sequence terminates in a grasp configuration. This narrowing in configuration space is mirrored by improvements in the precision of Robonaut’s estimated position of the box.

## REFERENCES

- [1] K. Chang, R. Holmberg, and O. Khatib, “The augmented object model: Cooperative manipulation and parallel mechanism dynamics,” in *Proc. IEEE Int’l Conf. on Robotics and Automation*, 2000.
- [2] J. Tan, N. Xi, and Y. Wang, “Integrated task planning and control for mobile manipulators,” *Int’l Journal of Robotics Research*, vol. 22, no. 5, 2003.
- [3] D. MacKenzie and R. Arkin, “Behavior-based mobile manipulation for drum sampling,” in *Proc. IEEE Int’l Conf. on Robotics and Automation*, 1996.

- [4] L. Petersson and H. Christensen, “A framework for mobile manipulation,” in *7th International Symposium on Robotics Systems*, 1999.
- [5] B. Pimentel, G. Pereira, and M. Campos, “On the development of cooperative behavior-based mobile manipulators,” in *Proc. of the Int’l Conf. on Autonomous Agents*, 2002.
- [6] R. Burridge, A. Rizzi, and D. Koditschek, “Sequential composition of dynamically dexterous robot behaviors,” *International Journal of Robotics Research*, vol. 18, no. 6, 1999.
- [7] R. Platt, A. Fagg, and R. Grupen, “Extending fingertip grasping to whole body grasping,” in *IEEE Int’l Conference on Robotics and Automation, Taipei, Taiwan*, May 2003.
- [8] M. Huber, “A hybrid architecture for adaptive robot control,” Ph.D. dissertation, U. Massachusetts, 2000.
- [9] Y. Nakamura, *Advanced Robotics Redundancy and Optimization*. Addison-Wesley, 1991.
- [10] M. Huber and R. Grupen, “Robust finger gaits from closed-loop controllers,” in *IEEE Int’l Conf. Robotics Automation*, 2002.