PR2: Opening Spring-Loaded Doors

Steven Gray, Christopher Clingerman, Maxim Likhachev, and Sachin Chitta

Fig. 1. The PR2 pushing open a spring-loaded door. Once the base is in contact with the door, the arm can let go of the door and use the base to keep pushing the door open as it moves through.

I. INTRODUCTION

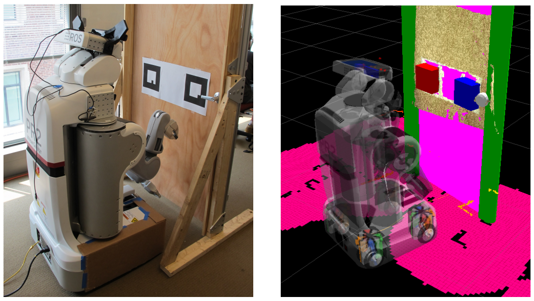
Opening doors is necessary in order to enhance and expand the set of tasks an automated robot can perform indoors. Autonomously and dynamically planning for opening both spring- and non-spring-loaded doors is essential to provide this functionality. Doors vary greatly with respect to size, shape, space constraints, and handles; therefore, hard-coded and precomputed motions designed to open doors can easily fail when designing a robust system. In order to build such a system on a mobile manipulation platform, one must tackle several problems, such as detecting the type of door and location of the handle, autonomously reaching for and grasping the handle, and coordinating the arm and base of the robot to open the door with respect to space constraints in the immediately surrounding area. Our work focuses mainly on the last of these issues.

Our approach uses a low-dimensional, graph-based representation of the problem in order to plan a door-opening procedure quickly and reliably. Due to this representation we can expect several advantages, such as robust support for various doors and their surrounding environments as well as utilization of the vast array of research that has been done with respect to heuristic search-based planning. In the extension to spring-loaded doors, planning is much more important due to the need to prevent the door from closing when not actively opening the door. Control must be transferred very carefully to open and maintain the position of the door.

The bulk of our work is an extension of [1]. Other research has been conducted in the past with respect to door opening using mobile manipulators, and many of these efforts are mentioned in [1]. The contribution of this paper is to add spring-loaded doors to the set of doors that can successfully be planned for and opened reliably. This addition requires extensive modifications to the existing door planner. In particular, these modifications pertain to the state space and the set of acceptable successor states.

II. SYSTEM ARCHITECTURE

The door-opening task consists of two main stages: first detecting the door and grasping the handle, followed by planning and executing the door-opening motion. For determining the door and handle sizes and positions, we use *a priori* knowledge of the door being used either in simulation or real-life trials. However, the initial position and relative open angle are determined using AR Toolkit wrapped in a ROS package [2], tracking two fiducial markers. Not only do these assist the robot in the initial approach, but they also help keep track of the progress of the main door-opening action.



(a) (b)

Fig. 2. AR Toolkit detection of the door using the wide-stereo cameras (a) and the representation of the door in rviz (b).

Planning and opening a door requires several key steps to ensure reliability in diverse situations. These steps are: (a) moving the hand into position around the handle and closing the gripper; (b) turning the handle to unlatch the door; (c) determining from the known model of the door whether pushing or pulling is required; (d) planning the door-opening motion; and (e) executing the plan. These are the necessary elements for pushing and pulling open any type of door. However, when moving from non-spring-loaded to spring-loaded doors, additional concerns must be addressed.

S. Gray and C. Clingerman are with the GRASP Laboratory, University of Pennsylvania, USA `{stgray, chcl}@seas.upenn.edu`

M. Likhachev is with the Robotics Insitute, Carnegie Mellon University, USA `maxim@cs.cmu.edu`

S. Chitta is with Willow Garage, Menlo Park, 94025, USA `sachinc@willowgarage.com`

It is now necessary to hold the door open in some sense at all times. If we require switching arms or which handle is being held, as is likely the case when pulling open the door, the base must be moved into contact with the door (or the other arm used to support the door) before releasing the handle. These additional constraints are incorporated into the planning stage; the planner realizes the switch is necessary in order to completely open the door.

III. MOTION PLANNING

If the planner were to take into account the three degrees of freedom for the planar base motion as well as the seven degrees of freedom for either arm, finding graph-search solutions for door opening would be computationally prohibitive. Due to the requirement, whenever the hand is grasping a handle, that the base be no farther from the door than the length of the arm, we can easily reduce the search complexity of the arm by substituting an inverse kinematics solution in its place. With the addition of spring-loaded doors, this inverse kinematics constraint is supplemented with a torque constraint for the door, limiting the allowable arm configurations and motions. The base of the robot must either be in contact with the door, or it must move such that the gripper is able to stay on the door handle, and the configuration of the arm must allow sufficient force to be exerted normal to the door to push or pull it open. The final goal for the base is not specified; the goal of the planner is to fully open the door.

The planner generates a lattice of motion primitives for the base and then finds the lowest cost path using Anytime Repairing A* (ARA*) [3]. That is, for each state we expand, we determine whether a certain motion primitive is feasible and associate the resulting state with a cost. The cost function combines the distance of the base to obstacles and an arm-configuration cost. As in [1], we require (x, y, θ) to represent the position and orientation of the base on the plane. Instead of storing the door angle directly, a binary variable called the *door interval* is used. The door interval is 0 when the door is at an angle where it may be fully closed without colliding with the robot. A door interval of 1 denotes that the door is at an angle where it may be fully opened without colliding with the base. Note that if the PR2 is far enough from the door, these intervals overlap. Lastly, for spring-loaded doors, we also have a state variable to indicate which part of the robot is in contact with the door; this can be the right arm, left arm, or the base.

A. *Torque Constraints*

Possible door handle locations in the PR2 baselink frame can be described by three values as shown in Fig. 3. The angles α and θ represent the angles of the shoulder-gripper vector and the +x-axis gripper vector relative to the +x-axis of the base. The distance between shoulder and end effector is given by r. The door handle height is known and constant. The maximum force the arm can exert normal to the door (in the direction indicated by θ) can then be calculated; this value is precomputed for a range of (α, θ, r).

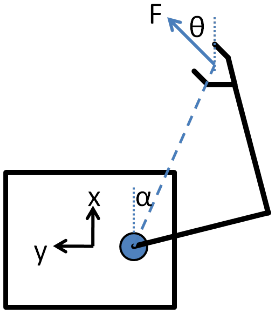


Fig. 3. Values used for the precomputation discretization. α and θ are the angles of the shoulder-handle vector and exerted force vector relative to the x-axis of the base, while r is the magnitude of the shoulder-handle vector.

The computation begins by solving the inverse kinematics. Because the PR2 arm is redundant, the inverse kinematics solver uses an initial guess for one of the joint angles. With the joint angles solved, the end-effector Jacobian is calculated, and can be plugged into the following relation:

$$\tau = J^T F \tag{1}$$

where τ is the vector of joint torques and F is the wrench applied at the end effector. F we specify as a unit force in the direction indicated by the angle θ. Because of the linear relationship between τ and θ, we can find the maximum force the arm in that configuration can exert on the door by multiplying by:

$$\min_{j \in joints} \frac{\tau_{max_j}}{\tau_j} \tag{2}$$

where the τ_{max} values are the maximum torques allowed at each arm joint. Because of the arm redundancy, this process is repeated for twenty unique inverse kinematic solutions, and the minimum force selected. In this manner, precomputation is done for each (α, θ, r). The precomputed values are read by the planner at startup; states where the door requires more force than the robot can exert from its position are disallowed.

B. *Switching Arms*

Aside from typical transitions between motion primitives for the base, the robot can also transition between which arm or base is in contact with the door. When either arm is on the door handle, the robot has two possible transitions: brace the opposite side of the door with the free arm, or move the base in contact with the door. From the base, the robot may push the door or transition to either arm. Transitions are allowed to occur when the valid door angle range for either arm overlaps with that of the base; the door angle range for the base is constrained such that the door must be within 5 cm of the base.

REFERENCES

[1] S. Chitta, B. Cohen, M. Likhachev, Planning for Autonomous Door Opening with a Mobile Manipulator, in *Proc. ICRA 2010.*

[2] G. Dumonteil, artoolkit package for ROS, `www.ros.org/wiki/artoolkit`, August 2011.

[3] M. Likhachev, G. Gordon, S. Thrun, ARA*: Anytime A* with provable bounds on sub-optimality, in *Proc. NIPS 2003.*