A Single Planner for a Composite Task of Approaching, Opening and Navigating through Non-spring and Spring-loaded Doors

Steven Gray, Sachin Chitta, Vijay Kumar, Maxim Likhachev

Abstract— **Opening and navigating through doors remains a challenging problem, particularly in cluttered environments and for spring-loaded doors. Passing through doors, especially spring-loaded doors, requires making and breaking contacts with the door and preventing the door from closing while passing through. In this work, we present a planning framework that handles non-spring and spring-loaded doors, in cluttered or confined workspaces, planning the approach to the door, pushing or pulling it open, and passing through. Because the problem is solved in a combined search space, the planner yields an overall least-cost path. The planner is able to insert a transition between robot-door contacts at any point along the plan. We utilize a compact graph-based representation of the problem to keep planning times low. We precompute the force workspace of the end-effectors to eliminate checks against joint torque limits at plan time. We have validated our solution in both simulation and real-world experiments on the PR2 mobile manipulation platform; the robot is able to successfully open a variety of spring-loaded and non-spring-loaded doors by pushing and pulling.**

I. INTRODUCTION

Opening doors is necessary in order to enhance and expand the set of tasks an autonomous robot can perform indoors. The majority of commercial doors are equipped with automatic mechanisms to ensure closure. These types of doors are typically called *spring-loaded* doors, whereas doors that do not close automatically are known as *non-spring-loaded*. Autonomously planning for opening both spring- and non-spring-loaded doors is essential to provide the functionality required of a useful indoor robot. One must tackle several problems in order to build an integrated door opening implementation for a mobile manipulation platform, such as detecting the type of door and the location of the handle, building a kinematic model of the door, and coordinating the arm and base of the robot to open the door with respect to space constraints in the immediately surrounding area. Our focus is on the last issue: *i.e.,* planning for and coordinating the motion of the arm and the base to approach, open, and pass through doors.

Although door opening in indoor environments has been widely addressed in recent work for mobile manipulation systems, ([1], [2], [3], [4], [5]), opening and moving through doors is still a challenging problem. Doors vary 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. Reactive approaches and low-level controllers may fail to consider obstacles and may need to be modified to handle doors with different parameters. Opening and navigating through doors, especially spring-loaded doors, requires making and breaking contacts with the door. For spring-loaded doors, the robot must maintain contact with the door and actively counteract the spring to keep it from closing. For doors in cluttered or confined environments, it is often necessary to switch the side of the door the robot contacts. We present, to the best of the authors' knowledge, the first planning framework that handles non-spring and spring-loaded doors, functions in cluttered or confined workspaces, and plans the approach to the door, pushing or pulling it open, and passing through. The plan is generated in a unified search space, including transitions between different robot-door contacts (for example, base against the door as well as gripper on the door handle), finding a least-cost solution for traversing doors. This is important because it allows the planner to decide the best location and time to transition from opening the door to moving through it. In contrast, disjoint approaches like hierarchical planners will specify but not modify the transitions.

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.

Our approach to motion planning starts with using a low-dimensional, graph-based representation of the problem in order to plan a door-opening procedure quickly and reliably. This provides several advantages including the ability to quickly plan for opening various doors and account for walls or other obstacles in their surrounding environments. If necessary, contact must be transferred very carefully between the arms and the base of the robot to open and maintain the

S. Gray and V. Kumar are with the GRASP Laboratory, University of Pennsylvania, Philadelphia, 19104, USA {stgray, chcl, kumar}@seas.upenn.edu

M. Likhachev is with the Robotics Insitute, Carnegie Mellon University, Pittsburgh, 15213, USA maxim@cs.cmu.edu

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

position of the door throughout execution of the opening action. Our planning algorithm can take into account which (if any) part of the robot is currently holding the door open and allows for transitions between arms as well as using the base to push against the door. We note that, particularly for pulling spring-loaded doors, it is very difficult to open the door and maneuver such that the base is in position to hold the door open as it passes through the door frame. Our planner handles this difficult situation.

A robotics platform for use in door opening is greatly aided by having some method for manipulating the door from both sides. For the purposes of this work, we assume a dual-arm mobile manipulator is used. Our approach is validated by an extensive set of experiments performed using the PR2 robot, a platform used extensively for navigating and acting within indoor environments ([6],[7]). The experiments involved multiple tests for opening a variety of doors (pulling and pushing both spring- and non-spring-loaded doors).

II. RELATED WORK

Robotic door opening has received a fair amount of attention in recent years. Within the overall task of robots opening doors, research can involve visual identification of doors and door handles ([8],[9],[3],[10]) or the physical action of opening the door. We simplify door identification since our contribution relates to the planned robotic motion required for door opening; door detection is outside the scope of this work. We chose a simple visual identification system based on the ARToolKit [11], but any other can be used.

Recent work has seen a number of robotic platforms addressing door opening. Early experiments ([1],[2]) have led to a number of systems designed for this task. However, as we mentioned in [4], many of these systems have not completely solved the door opening problem. Some do not pull open doors ([12],[10],[3],[6]). Others such as [13] use impedance control to open doors while learning the kinematic model, but may hit obstacles and do not move the robot base.

Purely reactive approaches such as [14],[15] will not work in complex environments and for multiple contacts. Planning algorithms provide a way to take into account factors that may not have been considered when designing a completely precomputed action or a reactive controller. There are also planning approaches that do not include motion of the robot base while opening doors. [16] plans manipulation motions for the opening of cabinet doors, allowing switching between different caging grasps, but the base of the robot remains stationary. Task space regions can accommodate pose constraints [17], but have not been used to simultaneously plan arm and base trajectories for door opening.

There have been other recent approaches to door-opening systems. However, none of these combine all of the following: switching contacts to allow end-effectors or the body of the robot to brace the door; the ability to handle spring-loaded doors; and combining the approach to the door, including selecting an initial contact with the door, and opening/passing through the door as a single plan. A number of trajectory planners which take into account external wrenches on the end-effector have been developed both for wheeled platforms ([18], [19]) and humanoid platforms ([20], [21]). However, these methods only allow pushing doors and cannot switch contact locations. In [22], the authors create a behavior-based system able to push and pull doors open, but do not deal with spring-loaded doors and obstacles. In [23], the authors estimate door parameters and pull open a door with a modular re-configurable robot featuring passive and active joints, but cannot pass through spring-loaded doors. In [24], Dalibard *et al.* introduce random sampling planning algorithms for humanoid robots to move through a doorway while also opening and closing the door. They have demonstrated results using both arms of the robot to move through the door and avoid obstacles, but are unable to pass through spring-loaded doors. In [5], Jain and Kemp extend previous work in which a robot could successfully open doors and drawers to include motion of the robot base. Their work does not include switching contact locations and would not allow the robot to brace open and pass through a spring-loaded door.

We build on our prior work of [4], in which collision-free trajectories were generated for opening non-spring-loaded doors. The previous system was limited to a single contact, the end-effector upon the door handle. We build upon this by adding transitions between robot-door contacts, including planning for the initial contact with the door. Pushing and pulling spring-loaded doors are handled by incorporating additional constraints on maintaining contact with the door. State feasibility is determined by checking against a precomputed map of the force-workspace. In contrast to previous work, the entire plan from approach, to opening, to moving through doors is computed in a single search. This allows a least-cost solution to be found that ensures the feasibility of contact transitions. Unpublished results have been shown at two workshops ([25], [26]).

III. MOTION PLANNING

We solve the planning problem for approaching, pushing and pulling open both non-spring and spring-loaded doors, and moving through them. The approach is most beneficial for doors that cannot be fully opened while the base remains stationary, but is also useful for other doors in constrained spaces. When attached to the door, we constrain the motion of the manipulator to a 1D manifold traced by the path of door handle.

The door-opening problem is to find a configuration path such that:

1) the robot passes through the door
2) robot avoids self-collision and collision with obstacles
3) path is feasible with respect to kinematic constraints

Spring-loaded doors have additional constraints:

1) once contact is made, some part of the robot is in contact with the door at all times
2) keeping the door open cannot violate joint torque limits

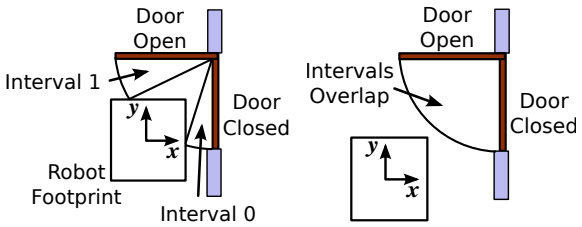


Fig. 2. Door intervals do not overlap when the robot intersects the area swept by the door (a), but do overlap as the robot moves further away (b).

Our planning algorithm operates by constructing and searching a graph of predefined and dynamically generated motion primitives [27]. The graph search uses the constructed graph to find a path from the start state (corresponding to the current position of the robot with respect to the door and the current door angle) to *any* state satisfying the goal conditions, specifically opening the door such that the robot can pass through the door frame by moving forward.

In the following sections, we explain the algorithm, covering the state-space representation, motion primitives, cost function and heuristics, and graph search.

A. Graph Representation

The graph is constructed using a lattice-based representation. A lattice is a discretization of the configuration space into a set of states and connections between those states, where every connection represents a feasible path. Let $G = (S, E)$ denote the graph G we construct, with S the set of states and E the set of transitions between states. To discuss the states in S, let us first consider the motion of a mobile manipulator opening a door. Let $(x_b, y_b, \theta_b) \subset SE(2)$, where θ_b is the heading, represent the configuration of the base, and $\theta_d \subset \mathbb{R}$ the set of possible door angles. One additional variable is needed to store the free angle of the manipulator. This produces states of 5 continuous variables. Storing the side of the door the robot is contacting, as well as the part of the robot in contact with the door, takes additional, though discrete, variables. We consider right end-effector on handle, left end-effector on handle, and base against door contacts.

As we mentioned in [4], it is sufficient to use a more compact representation of the door angle. Instead of storing the door angle θ_d directly, we utilize a discrete variable, d, called the *door interval*. Door intervals are illustrated in Figure 2. 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. The two intervals are separate if the body of the robot intersects the swept area of the door, as is the case in Figure 2a. If the robot is far enough from the door, as in Figure 2b, these intervals overlap, denoted with a value of 2. We note that, though the door angle is not stored, the planner has the ability to quickly reconstruct the set of door angles for a robot pose p, $\Lambda(p)$, feasible given the current contacts.

Additionally, instead of storing the free angle of the manipulator, it is sufficient to place restrictions on the manipulator. We chose to restrict it to elbow-down configurations. Conservative collision checking can be done against the swept volume of the arm, which can be precomputed for end-effector poses. In our compact representation, a state in the state-space used by the planner, $s \in S$, is given by

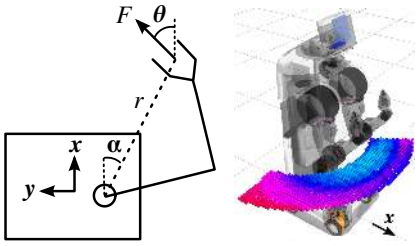
$$s = (x_b, y_b, \theta_b, d, h, v)$$

where d is the door interval, h is the side of the door whose handle is being grasped, and v indicates the part of the robot in contact with the door. v takes 4 possible values, corresponding to no contact, left end-effector on door handle, right end-effector on door handle, and base against door.

We use a lattice-based planning representation ([28], [29]) to define the set of transitions E between states. A motion primitive is a discretized, finite-length feasible path between neighboring states. It can be defined as a discretized path of intermediate offsets of (x_b, y_b, θ_b) and transitions in d, h, v, or some subset thereof. The lattice graph is dynamically constructed by the graph search as it expands states.

We use two different types of motion primitives that connect a state s to a successor state, $s' \in succ(s)$. The first primitives describe motions for the mobile base. For a holonomic base they represent forward and backward translational motion, rotation in place, strafing, and moving forward and backward while turning. For a nonholonomic base, they satisfy the nonholonomic constraints on its motion. These primitives are augmented with transitions between door interval values d, generated at runtime because changes in d are a result of moving the base with respect to the door. In particular, when moving the base from a pose that is completely outside the swept area of the door (as in Figure 2b) to a pose within the swept area (as in Figure 2a), there are two copies of the states being created; one with $d = 1$ and one with $d = 0$. The second set of primitives do not include motion for the base. Instead, these are transitions between discrete variables representing the part of the robot in contact with the door v and the side of the door being grasped h. In other words, these transitions correspond to such actions as removing the end-effector from the door handle and bracing the door with the robot base, or grasping the door handle on the opposite side of the door with the free end-effector and releasing the currently held door handle. Because these primitives do not include motion of the mobile base, they cannot transition between values for the door interval d.

Before a successor of state s, s' can be added to the lattice graph, it must first be checked for feasibility. For a successor to be valid, for every pose p along the discretized motion primitive, the set of valid door angles $\Lambda(p)$ must overlap between adjacent poses. The corresponding door intervals for adjacent poses must also be the same (or include the overlapping door interval, $d = 2$). This way, the base can move along the motion primitives from one pose to another while continuously contacting the door. Additionally, the robot base and conservative arm estimate must be collision free. The allowed contact transitions for the planner are as follows: (1) when the robot is not yet in contact with the door, it may contact the door with either arm or the base;



(a) (b)

Fig. 3. (a) 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. (b) Force workspace computation for the right arm, (for $\alpha \in [-1.9, 0.6]$ and $r \in [0.45, 0.85]$) with color representing the minimum normal force the arm can apply in a given direction, with light blue near the center containing the greatest force at 34.4N and red at the sides being the least. Values shown for $\theta = 0$.

(2) when an end-effector is on a handle, the planner may transition to using the other arm on the other side of the door, or bring the base into contact with the door, provided the valid angles for old and new contact overlap. If the robot is not yet in contact with the door, the only valid angle is the door closed angle. Additionally, if the door is to come into contact with the base, the valid door angles are such that the door is no more than 5 cm from the base.

1) Precomputation: Finding the valid door angle ranges $\Lambda(s)$ for a given state s can be expensive, motivating moving as much computation as possible off-line. To check whether a given door angle is valid requires checking for valid inverse kinematics solutions or checking that the end-effector pose is within the (precomputed) robot's reachable workspace. Further, for spring-loaded doors, the ability to exert a given force normal to the door can be checked by referencing the robot's force workspace, similar to force workspace approach presented in [30]. The force workspace precomputation also functions as a reachable workspace precomputation, returning the reachable workspace when the query is set to regions where the allowable force is greater than zero. For our purposes, the force workspace precomputation only has to be done for a horizontal plane at the height of the door handle. It is worth noting that we assume that the door is moving relatively slowly so the quasi-static assumption is appropriate.

We precompute the force workspace values for a range of robot positions and door angles. Possible door handle locations in the robot base frame can be described by three values as shown in Figure 3a. The angles α and θ represent the angles of the shoulder-gripper vector and the gripper applied force relative to the x-axis of the base. The distance between shoulder and end-effector is given by r. For a given inverse kinematics solution for the arm, (specifying a value for the free angle) we can calculate the maximum force the arm is able to exert normal to the door in the θ direction. The end-effector Jacobian is calculated and then plugged into the following relation:

$$\tau = J^T F_e$$

where τ is the vector of joint torques and F_e is the wrench applied at the end-effector. Referring to Figure 3, F is specified as the force component of the wrench F_e in the direction indicated by the angle θ in a plane parallel to the ground plane.

We exploit the linear relationship between τ and F to scale the value of F by $\min_{i \in joints} \tau_{i,MAX}/\tau_i$ to get the maximum allowed force for that configuration. Because the arm is redundant and multiple inverse kinematics solutions exist for a given desired end-effector pose, this process is repeated for twenty elbow-down IK solutions and the minimum force across all these solutions is selected. If no IK solution exists, we record the allowed force as zero. Values for $\theta = 0$ are shown in Figure 3b.

B. Cost Function and Heuristic

The cost of a transition in our graph representation is defined as the sum of the two terms,

$$c(s, s') = c_{movement} \times c_{costmap} + c_{door}$$

In the first term, $c_{movement}$ is the cost associated with moving the robot along a given base motion primitive. This term depends on the time it takes to execute the motion primitive. It also allows the user to minimize the use of certain primitives, such as moving backward. $c_{costmap}$ is given by using the maximum cost of the 2D costmap cells traced by the robot footprint along the transition. The costmap projects world obstacles to a 2D grid and represents proximity to obstacles with increased costs. The second term, c_{door}, is proportional to the change in door angle nearest to the center of the robot's reachable workspace from state to state. To enable this, for each state we record a door angle deemed $\lambda(s)$ which minimizes a function punishing deviation from a nominal shoulder-handle distance r_c and angle α_c:

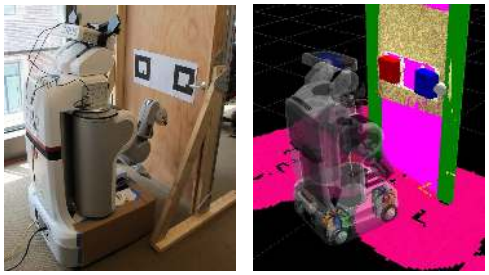
$$\lambda(s) = \min_{\theta_d \in \Lambda(s)} \left(1 - \frac{1}{1 + (r - r_c)^2(\alpha - \alpha_c)^2}\right)$$

where r and α are defined in Figure 3 and r_c and α_c correspond to the center of the reachable workspace for the arm being used. Of course, if the robot has not yet contacted the door or the base is in contact, this term is ignored. Transitions associated with switching between robot-base contacts have fixed costs determined by the user, allowing the user to penalize switching if desired.

The purpose of the heuristic is to efficiently guide the search towards a feasible solution. Since part of the condition for a state s to be considered a goal state is that $\Lambda(s)$ overlaps with the fully open door angles, we set the first term of the heuristic to estimate the remaining angle the door needs to be opened before it is considered fully open. The second term in the heuristic is a term for the distance of the robot to a circle of radius r_{len} around the door. The heuristic is given as:

$$H(s) = \max(0, |\mathbf{x}_{robot} - \mathbf{x}_{hinge}| - r_{len}) + |\lambda(s) - \lambda_{open}|$$

We set r_{len} to the length of the door from hinge to handle plus the length of the arm. The purpose of this term is to estimate the cost of the plan to drive to the door and make



(a) (b)

Fig. 4. ARToolKit detection of the door using one of the wide-stereo cameras (a) and a visualization of the door (b). The inflated 2D costmap is in pink.

contact. Both of these terms are admissible and consistent. When the robot is further than r_{len} from the door, and thus v is empty because the robot is not in contact with the door, the first term remains at its maximum value. When the robot is able to contact the door (v is nonempty), the second term in the heuristic is zero and the first term may decrease. Because of this, the sum of the terms is an admissible and consistent heuristic.

C. Search

Given a graph as defined above, composed of states linked by motion primitives, we need an efficient way to search it for a solution path. A* search is a very popular method for graph search that finds an optimal path, which may not be possible if deliberation time is limited [31]. Instead, we use an anytime variant, Anytime Repairing A* (ARA*) [32]. The algorithm generates an initial, possibly suboptimal solution then focuses on improving the solution while time remains. The algorithm is provably complete for a given graph G and provides theoretical bounds on sub-optimality of solutions. It works by inflating the heuristic by a value $\epsilon \geq 1$. Given additional time, the graph search is able to decrease the bound ϵ to 1.0 and provide the optimal solution.

IV. IMPLEMENTATION

The door-opening task consists of two main stages: first detecting the door and determining its parameters, followed by planning and executing the door-opening motion. Door detection is outside the scope of this work; 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. The initial position and relative open angle are determined using the ARToolKit [11], which provides a framework for tracking fiducial markers. Each door is measured in advance and several door properties are recorded, including the distance from each marker to the edge of the door, the distance to the door handle, the depth of the door handle, the side with the hinge, the direction of swing, and the necessary force at the handle to open the door. We make the assumption that the required force remains constant throughout the trajectory, though [33] shows that the required force diminishes as the door opens.

Our testbed is a Willow Garage PR2, as shown in Figure 4. The robot has two arms with 7 degrees of freedom, an omni-directional base, a pan-tilt head, and an adjustable height torso. We use a Hokuyo scanning laser range finder attached to the base and a tilting laser scanner to generate a 3D collision map and 2D costmap for navigation. The left camera of the wide stereo-camera pair is used to detect the ARToolKit markers.

The planner yields a trajectory of n states of the form $s = (x_b, y_b, \theta_b, d, h, v)$. Before the path can be executed on the robot, the inverse kinematics at each position must be resolved, requiring the door angles. Because for each state there may be many feasible door angles in a given door interval, it is desirable to minimize the motion of the door. We formulate and solve the following quadratic problem as a post-processing step:

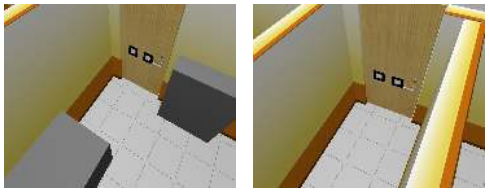
$$\begin{array}{c} \min \sum_{i \leq n} ||\theta_i - \theta_{i-1}||_2^2 \\ s.t.\ \theta_{i,LB} < \theta_i < \theta_{i,UB} \end{array}$$

solving for the door angle at each step i, where the upper and lower bounds are given by maximum and minimum angles in $\Lambda(s_i)$. With the door angles known, the door handle locations and thus end-effector poses are known. Combined with the trajectory for the base, we have sufficient information to generate the joint space trajectory for the arms by solving the IK for the arm at each step (or at least those steps in which an arm is attached to the door). Because of the force workspace precomputation and conservative collision checking, we know an IK solution exists. To resolve the kinematic redundancy of the arm, the IK solver uses an initial seed value for one of the joint angles, then analytically solves the remaining 6 degrees of freedom. In some situations, such as an interior corner door, the presence of walls may separate inverse kinematics solutions into disjoint sets (elbow up and elbow down). We handled this by only using elbow-down configurations. Transitions between arms involve calls to an arm-specific planner using sampling-based motion planning [34]. At this point, a trajectory for the base and joint angle trajectories for the arms are known.

V. EXPERIMENTAL RESULTS

We have tested our planner on a simulated PR2 and on the physical robot. For all of the experiments, the environments are discretized at a resolution of 2.5 cm. The robot base heading is discretized at 22.5 degrees; 13 motion primitives associated with motion of the base are given for each heading. Path lengths of the primitives range from 2.5 cm to 20 cm. Typical plans were between 20 and 110 motion primitives in length.

Two of our simulated environments are shown in Figure 5. The first set of simulated tests, with results in Tables I and II, shows how the number of states expanded and planning time vary between open and tight spaces. The corner hinge and corner edge environments refer to the narrow hallway environment of Figure 5b, minus the right or left wall, respectively. As for the office environment, Figure 5a, the obstacle along the left wall is placed at different distances to the door. The corner hinge case is relatively open and simple for the planner to solve; the robot may move through



(a) (b)

Fig. 5. Simulated environments include an office door with nearby obstacles (a) and a narrow hallway (b).

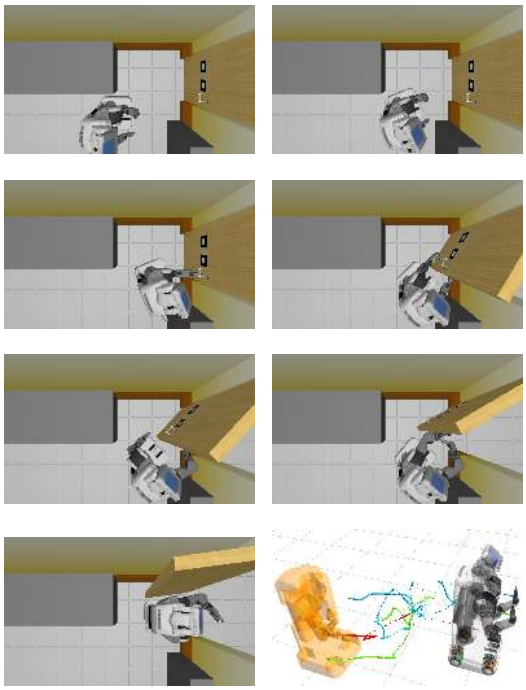


Fig. 6. Frames from a successful test in the office environment. The left obstacle is 1.2 m from the door frame. The last image shows a trace of the base and end-effector locations throughout the motion. Note from the base trajectory (red) that the robot backs up and turns as it initially pulls open the door.

the door as soon as the door open angle is large enough for it to pass through. However, a wall placed on the other side of the door as in the corner edge case requires the robot to open the door further before it can transition to holding the other side of the door and pass through. The narrow hallway is quick to solve because walls on both sides greatly limit the possible states. Conversely, when starting further from the door, as in the office door case shown in Figure 6, the fixed cost of transitions to contact the door lead to more states being expanded prior to the contact. With more room (as the obstacle is further and further away from the door) more and more states are expanded. The resulting paths are of decreasing cost as more direct routes from the start to the door become obstacle-free.

The second set of tests, shown in Table III, illustrates the effects of changing the force required to open spring-loaded doors. From the same start state, the plans for 1 N and 10 N are identical; joint torque limits were not a limiting factor at under 10 N. However, moving to 15 N, some of the transitions in the previous plans are infeasible so the search must expand more states to route around the infeasible regions. This leads to a longer, higher cost solution. For 20 N and above, no solution exists for pulling open doors; the arms of the PR2 are not strong enough to hold such doors open to allow transitioning from contacts on one side of the door to the other.

Door	Planning Time (s)		Expansions		Final Soln. Cost
	First Soln. $\epsilon = 5.0$	Final Soln. $\epsilon = 1.0$	First Soln. $\epsilon = 5.0$	Final Soln. $\epsilon = 1.0$	$\epsilon = 1.0$
Corner Hinge	0.97	2.23	2249	6838	551
Corner Edge	3.32	6.17	11267	25790	880
Narrow Hall	0.13	0.21	873	1160	822
Office (1.0m)	0.74	1.79	2148	6115	707
Office (1.2m)	0.80	2.13	2513	7512	707

TABLE I

SIMULATION: PLANNING TIMES, EXPANDED STATES, AND COSTS FOR **PULLING** DOORS OPEN.

Door	Planning Time (s)		Expansions		Final Soln. Cost
	First Soln. $\epsilon = 5.0$	Final Soln. $\epsilon = 1.0$	First Soln. $\epsilon = 5.0$	Final Soln. $\epsilon = 1.0$	$\epsilon = 1.0$
Corner Hinge	1.07	4.97	7468	31448	775
Corner Edge	4.06	8.79	26516	55134	1076
Narrow Hall	0.17	0.41	1347	1855	750
Office (0.8m)	3.93	6.69	21859	37937	2085
Office (1.0m)	5.03	9.69	26540	48897	1685
Office (1.2m)	6.03	11.11	32879	60032	1463

TABLE II

SIMULATION: PLANNING TIMES, EXPANDED STATES, AND COSTS FOR **PUSHING** DOORS OPEN.

Normal Force (N)	Planning Time (s)		Expansions		Final Soln. Cost
	First Soln. $\epsilon = 5.0$	Final Soln. $\epsilon = 1.0$	First Soln. $\epsilon = 5.0$	Final Soln. $\epsilon = 1.0$	$\epsilon = 1.0$
1.0	0.97	2.23	2249	6838	551
10.0	1.00	2.24	2249	6838	551
15.0	4.30	5.95	14948	24905	871

TABLE III

SIMULATION: PLANNING TIMES, EXPANDED STATES, AND COSTS FOR **PULLING** REQUIRING DIFFERENT NORMAL FORCES AT THE HANDLE.

Next, we discuss results on the physical PR2 for pushing and pulling both spring-loaded and non-spring-loaded doors. These results were gathered prior to a recent normalizing of the cost and heuristic functions and have much higher costs. Due to recent implementation optimizations, the previous results take more time per expansion. They also specify the initial contact with the door. The results of running twenty planning trials on a few doors for both pulling and pushing are listed in Table IV and Table V, respectively. The testbed (shown in Figure 4a) and conference room door were not spring-loaded. The kitchen door required 15 N normal force at the handle to open, while the office door required 27 N. In between trials, we moved the starting location of the base by a few centimeters and the initial orientation varied but was kept within ± 20 degrees of normal to the door. On average, both pushing and pulling plans took under 6 seconds to find an initial solution, in most cases, much less. The cost of pushing plans is higher, as the robot must plan

through a narrow passageway in the costmap, most likely with nonzero costs. Pulling plans allow the robot to withdraw from the door into open space of the costmap. The testbed door was also narrower than the kitchen door, requiring the robot to pass through higher cost cells in the costmap, but reducing the number of expansions (states added to the graph during planning) and thus yielding faster planning times. The conference room door, the longest to plan, bordered directly on a wall.

Images from the PR2 pushing open a spring-loaded door and pulling a non-spring-loaded door are shown in Figure 7. The video accompanying this paper contains these runs as well as pushing open an office door and pulling open the kitchen door. Note that the motion in the video is not smooth because we are commanding one waypoint for the arms and base at a time in order to keep them synchronized. Twenty trials total were carried out past the planning stage. The gripper managed to slide off the handle when pushing the kitchen door open, but the robot was able to successfully pass through the door; these trials have been counted as successes. As long as the robot was initially able to grasp the door handle (i.e., aside from door detection issues), the robot never failed to pull open a door which required 15 N or less normal force at the handle, and only once failed to push a door requiring less than 27 N of normal force at the handle. The one failure, due to an issue with the costmap, generated a path which collided with the door frame.

Failures outside the scope of the planning occurred due to poorly estimated door parameters. Such errors resulted in missed grasps of the door handle, as happened in eight additional trials. Errors in handle and hinge detection can generate large internal forces in the arm during the motion; the end-effector slipped off the door handle in three successful trials. Future work will incorporate door model estimation and replanning to improve robustness.

VI. CONCLUSION

In conclusion, we have implemented a graph-based search algorithm that allows a mobile manipulator, such as the PR2, to open both spring-loaded and non-spring-loaded doors. To the authors' knowledge, this is the first paper to incorporate switching between arms and the base to accomplish this task and demonstrate the results outside of simulation. Solutions to this planning algorithm are constrained to avoid obstacles as well as move the base of the robot and keep an arm in reach of the handle or the base against the door. The compact graph-based representation greatly reduces the dimensionality of the constrained planning problem, but requires an offline precomputation step for evaluating end-effector forces for a range of robot positions and door angles. We have verified the effectiveness of our approach with real-world experiments and can attest to the contribution our algorithm makes toward autonomous indoor navigation.

While applied to door opening, this work can be generalized to a variety of systems involving closed chains as mentioned in [35]. It is particularly relevant to systems in which making and breaking contacts with the environment will result in changes in chain topology, for instance, planning for humanoid robots navigating complex environments using both their arms and legs.

Door (Force, N)	Planning Time (s)		Expansions		Final Soln. Cost
	First Soln. $\epsilon = 5.0$	Final Soln. $\epsilon = 1.0$	First Soln. $\epsilon = 5.0$	Final Soln. $\epsilon = 1.0$	$\epsilon = 1.0$
Testbed (0)	0.249 ± 0.201	0.485 ± 0.244	79.8 ± 75.0	99.3 ± 68.6	135,340 $\pm 141,760$
Kitchen (15)	2.22 ± 1.59	2.95 ± 2.10	370 ± 265	465 ± 347	$12,841$ $\pm 7,683$
Conference (0)	3.50 ± 1.06	5.59 ± 2.76	601 ± 175	942 ± 486	$29,032$ $\pm 21,696$

TABLE IV

PLANNING TIMES, EXPANDED STATES, AND COSTS FOR **PULLING** DOORS OPEN. CONTAINS AVERAGES AND STANDARD DEVIATIONS FOR 20 TRIALS ON EACH DOOR.

Door (Force, N)	Planning Time (s)		Expansions		Final Soln. Cost
	First Soln. $\epsilon = 5.0$	Final Soln. $\epsilon = 1.0$	First Soln. $\epsilon = 5.0$	Final Soln. $\epsilon = 1.0$	$\epsilon = 1.0$
Testbed (0)	0.292 ± 0.069	2.04 ± 0.467	86.2 ± 20.8	649 ± 162	418,310 $\pm 38,174$
Kitchen (15)	2.20 ± 0.301	3.03 ± 0.245	916 ± 116	$1,228$ ± 80.9	$94,809$ $\pm 70,150$
Office (27)	0.636 ± 0.132	1.92 ± 0.251	195 ± 41.6	589.6 ± 91.8	$441,890$ $\pm 61,484$

TABLE V

PLANNING TIMES, EXPANDED STATES, AND COSTS FOR **PUSHING** DOORS OPEN. CONTAINS AVERAGES AND STANDARD DEVIATIONS FOR 20 TRIALS ON EACH DOOR.

VII. ACKNOWLEDGMENTS

We thank Willow Garage for their partial support of this work. In addition, this research was partially sponsored by the Army Research Laboratory Cooperative Agreement Number W911NF-10-2-0016 and the Office of Naval Research grants N00014-09-1-1031 and N00014-07-1-0829.

REFERENCES

[1] K. Nagatani and S. Yuta, "An experiment on opening-door-behavior by an autonomous mobile robot with a manipulator," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 1995, pp. 45–50.

[2] G. Niemeyer and J. Slotine, "A simple strategy for opening an unknown door," in *IEEE International Conference on Robotics and Automation (ICRA)*, 1997.

[3] E. Klingbeil, A. Saxena, and A. Y. Ng, "Learning to open new doors," in *Robotics: Science and Systems (RSS) Workshop on Robot Manipulation: Intelligence in Human Environments*, 2008.

[4] S. Chitta, B. Cohen, and M. Likhachev, "Planning for autonomous door opening with a mobile manipulator," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2010.

[5] A. Jain and C. C. Kemp, "Pulling open doors and drawers: Coordinating an omni-directional base and a compliant arm with equilibrium point control," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2010, pp. 1807–1814.

[6] W. Meeussen, M. Wise, S. Glaser, S. Chitta, C. McGann, P. Mihelich, E. Marder-Eppstein, M. Muja, V. Eruhimov, T. Foote, J. Hsu, R. B. Rusu, B. Marthi, G. Bradski, K. Konolige, B. Gerkey, and E. Berger, "Autonomous door opening and plugging in using a personal robot," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2010.

[7] E. Marder-Eppstein, E. Berger, T. Foote, B. Gerkey, and K. Konolige, "The office marathon: Robust navigation in an indoor office environment," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2010.

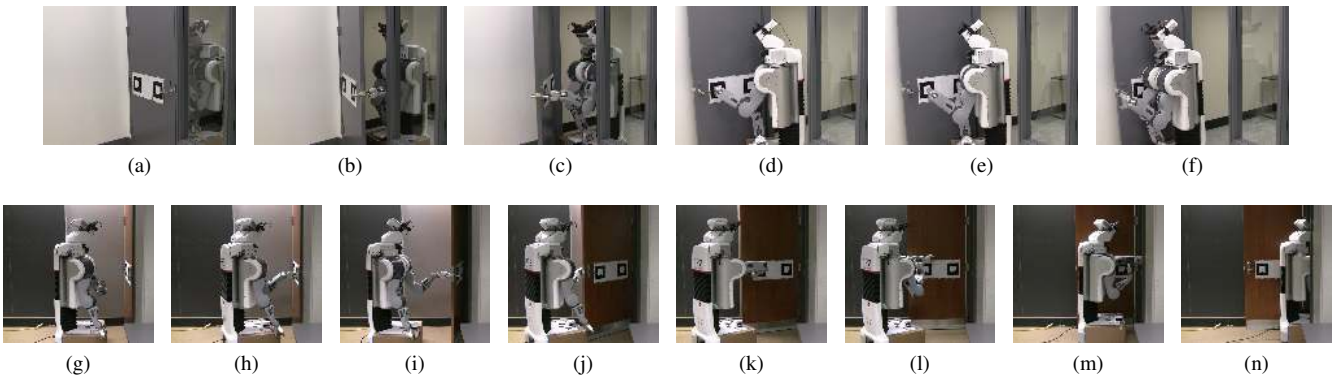


Fig. 7. The top row is an image sequence from pushing open a kitchen door requiring 15N normal force at the handle (a-f). Initially the left arm is on the door handle (a-d), then the plan transitions to pushing the door with the base (e-f). The bottom row is an image sequence from pulling open a non-spring-loaded door (g-n). The robot is initially not contacting the door (g), makes contact with the left arm and begins pulling the door open (h-j), transitions to contact using the right arm (k), then transitions to bracing with the left arm (l-m) and then base as it moves through the door (n).

[8] D. Anguelov, D. Koller, E. Parker, and S. Thrun, "Detecting and modeling doors with mobile robots," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2004.

[9] E. P. L. Aude, E. P. Lopes, C. S. Aguiar, and M. F. Martins, "Door crossing and state identification using robotic vision," in *8th International IFAC Symposium on Robot Control (Syroco 2006)*, September 2006.

[10] C. Ott, B. Bäuml, C. Borst, and G. Hirzinger, "Autonomous opening of a door with a mobile manipulator: A case study," in *IFAC Symposium on Intelligent Autonomous Vehicles (IAV)*, 2007.

[11] G. Dumonteil, "ARToolKit package for ROS," www.ros.org/wiki/artoolkit, August 2011.

[12] C. Ott, B. Bäuml, C. Borst, and G. Hirzinger, "Employing cartesian impedance control for the opening of a door: A case study in mobile manipulation," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) Workshop on Mobile Manipulators: Basic Techniques, New Trends & Applications*, 2005.

[13] T. Ruhr, J. Sturm, D. Pangercic, M. Beetz, and D. Cremers, "A generalized framework for opening doors and drawers in kitchen environments," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2012.

[14] L. Petersson, D. Austin, and D. Kragic, "High-level control of a mobile manipulator for door opening," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, vol. 3, 2000, pp. 2333–2338.

[15] C. Rhee, W. Chung, M. Kim, Y. Shim, and H. Lee, "Door opening control using the multi-fingered robotic hand for the indoor service robot," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2004.

[16] R. Diankov, S. Srinivasa, D. Ferguson, and J. Kuffner, "Manipulation planning with caging grasps," in *IEEE-RAS International Conference on Humanoid Robots (HUMANOIDS 2008)*, December 2008.

[17] D. Berenson, S. Srinivasa, and J. Kuffner, "Task space regions: A framework for pose-constrained manipulation planning," *International Journal of Robotics Research (IJRR)*, March 2011.

[18] J.-H. Kang, C.-S. Hwang, and G. T. Park, "A simple control method for opening a door with mobile manipulator," in *International Conference on Control, Automation and Systems (ICCAS)*, 2003.

[19] J. P. Puga and L. E. Chiang, "Optimal trajectory planning for a redundant mobile manipulator with non-holonomic constraints performing push-pull tasks," *Robotica*, vol. 26, pp. 385–394, May 2008.

[20] H. Arisumi, J.-R. Chardonnet, and K. Yokoi, "Whole-body motion of a humanoid robot for passing through a door - opening a door by impulsive force -," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2009, pp. 428–434.

[21] G. Digioia, H. Arisumi, and K. Yokoi, "Trajectory planner for a humanoid robot passing through a door," in *IEEE-RAS International Conference on Humanoid Robots (HUMANOIDS 2009)*, December 2009, pp. 134–141.

[22] B. J. W. Waarsing, M. Nuttin, and H. V. Brussel, "Behaviour-based mobile manipulation: The opening of a door," in *International Workshop on Advances in Service Robotics (ASER)*, 2003, pp. 168–175.

[23] S. Ahmad and G. Liu, "A door opening method by modular reconfigurable robot with joints working on passive and active modes," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2010, pp. 1480–1485.

[24] S. Dalibard, A. Nakhaei, F. Lamiraux, and J.-P. Laumond, "Manipulation of documented objects by a walking humanoid robot," in *IEEE-RAS International Conference on Humanoid Robots (HUMANOIDS 2010)*, 2010, pp. 518–523.

[25] S. Gray, C. Clingerman, S. Chitta, and M. Likhachev, "PR2: Opening spring-loaded doors," in *IEEE/RSJ International Conference on Intelligent Robots and Systems, PR2 Workshop*, 2011.

[26] S. Gray, C. Clingerman, S. Chitta, V. Kumar, and M. Likhachev, "Search-based planning for autonomous spring-loaded door opening," in *Robotics: Science and Systems (RSS) Workshop on Mobile Manipulation: Generating Robot Motion for Contact with the World*, 2012.

[27] B. Cohen, S. Chitta, and M. Likhachev, "Search-based planning for manipulation with motion primitives," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2010.

[28] M. Pivtoraiko and A. Kelly, "Generating near minimal spanning control sets for constrained motion planning in discrete state spaces," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2005, pp. 3231–3237.

[29] M. Likhachev and D. Ferguson, "Planning long dynamically-feasible maneuvers for autonomous vehicles," *International Journal of Robotics Research*, 2009.

[30] A. Madhani and S. Dubowsky, "Motion planning of mobile multi-limb robotic systems subject to force and friction constraints," in *Robotics and Automation, 1992. Proceedings., 1992 IEEE International Conference on*, may 1992, pp. 233–239 vol.1.

[31] P. E. Hart, N. J. Nilsson, and B. Raphael, "A formal basis for the heuristic determination of minimum cost paths," *IEEE Transactions on Systems, Science, and Cybernetics*, vol. SSC-4, no. 2, pp. 100–107, 1968.

[32] M. Likhachev, G. Gordon, and S. Thrun, "ARA*: Anytime A* with provable bounds on sub-optimality," in *Advances in Neural Information Processing Systems (NIPS)*. Cambridge, MA: MIT Press, 2003.

[33] A. Jain, H. Nguyen, M. Rath, J. Okerman, and C. C. Kemp, "The complex structure of simple devices: A survey of trajectories and forces that open doors and drawers," in *IEEE RAS and EMBS International Conference on Biomedical Robotics and Biomechatronics (BioRob)*, September 2010, pp. 184–190.

[34] I. A. Şucan, M. Moll, and L. E. Kavraki, "The Open Motion Planning Library," *IEEE Robotics & Automation Magazine*, 2012, to appear. [Online]. Available: http://ompl.kavrakilab.org

[35] S. Gray, C. Clingerman, V. Kumar, and M. Likhachev, "Motion primitive-based graph planning for mobile manipulation with closed-chain systems," University of Pennsylvania, Philadelphia, Pennsylvania, Tech. Rep. MS-CIS-12-06, March 2012. [Online]. Available: http://repository.upenn.edu/cis_reports/968/