Cart Pushing with a Mobile Manipulation System: Towards Navigation with Moveable Objects

Jonathan Scholz, Sachin Chitta, Bhaskara Marthi, Maxim Likhachev

***Abstract*— Robust navigation in cluttered environments has been well addressed for mobile robotic platforms, but the problem of navigating with a moveable object like a cart has not been widely examined. In this work, we present a planning and control approach to navigation of a humanoid robot while pushing a cart. We show how immediate information about the environment can be integrated into this approach to achieve safer navigation in the presence of dynamic obstacles. We demonstrate the robustness of our approach through long-running experiments with the PR2 mobile manipulation robot in a typical indoor office environment, where the robot faced narrow and high-traffic passageways with very limited clearance.**

I. INTRODUCTION

Navigating safely while manipulating moveable objects is a hard task especially in cluttered, crowded environments where the amount of space available is limited and the obstacles are non-stationary. Particularly in tasks like retail stocking or table cleanup, the robot may need to manipulate and carry a large number of objects. We address this need by describing a system for smoothly and safely pushing carts in such environments. This ability underlies not only navigation with carts, but with many standalone mobile objects such as chairs, tables, and boxes.

Robust mobile manipulation has been a topic of research for a long time but there are few examples of real-world applications. This shortage can be partially blamed on complexity: mobile manipulation requires a careful synthesis of navigation and manipulation capabilities. Navigating effectively requires consistent knowledge of the environment around the system, often achieved through vision, laser, proprioceptive and other types of sensors and an ability to respond to sudden changes in the environment. Effecting the environment, e.g. opening doors, often requires dealing with constraints that, if violated, can results in large internal forces. Mobile manipulation tasks like cart pushing will thus require a tight integration of several components including sensing, motion planning and control.

In this paper we present our approach to planning, control, and sensing for navigation with moveable objects, with a particular application to pushing holonomic utility carts. Our approach exploits a proven anytime planner for generating near-optimal plans given a set of motion primitives that can drive the robot and the cart. We also provide a local controller which functions in the robot odometric frame and is responsible for both following the plan and avoiding static and dynamic obstacles observed by sensors. Using a combination of these controllers and a 3D sensing framework, we are able to demonstrate reliable and robust navigation with a holonomic cart in a typical office environment using the PR2 mobile manipulation robot. Our work builds on previous successes in both navigation and manipulation on the PR2, and leverages the Robot Operating System (ROS) framework for integration of the multiple components needed for robust mobile manipulation [8], [11].

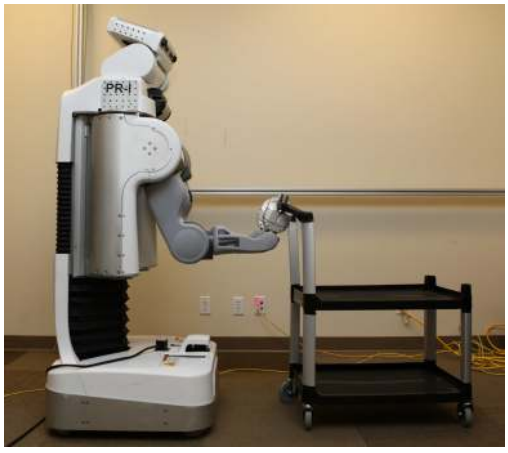


Fig. 1. The PR2 mobile manipulation robot with a holonomic cart

A. Related Work

The earliest results in pushing carts using robots were achieved using a single manipulator mounted on a mobile holonomic base [14], [15], [16]. In these systems the manipulator came into contact with the cart at a single point, and the problem was to solve for the effector forces required to produce desired trajectories with the cart. This work showed progress towards task-level cart manipulation, but was limited to tracing simple open-loop paths with the cart [16]. In contrast, our solution can execute smooth, arbitrary trajectories in a closed-loop controller using two arms.

Subsequent work explored the use of full humanoid robots for pushing carts with two arms. In principle, two arms can simplify the cart control problem by fully constraining all degrees of freedom. Honda's ASIMO is capable of pushing a cart while walking across a room and even up an incline,

Jonathan Scholz is a PhD student in Interactive Computing, Georgia Institute of Technology, 801 Atlantic Drive,Atlanta, GA 30332 jkscholz@gatech.edu

Sachin Chitta is a Research Scientist at Willow Garage Inc., 68 Willow Rd,. Menlo Park, CA 94025 sachinc@willowgarage.com

Bhaskara Marthi is a Research Scientist at Willow Garage Inc., 68 Willow Rd,. Menlo Park, CA 94025 bhaskara@willowgarage.com

Maxim Likhachev is a Research Assistant Professor at the Robotics Institute, Carnegie Mellon University, Pittsburgh, PA maximl@ri.cmu.edu

but does not make use of its arms to articulate the cart for navigation [13].

Several projects have explored the use of another humanoid robot, the HRP-2, for pushing mobile objects. One domain which shares our interest in manipulating large objects is *navigation among moveable obstacles* (NAMO). Although both domains involve navigation and manipulation, NAMO addresses the complementary problem of navigating the robot on a map containing obstacles that must be moved in order to achieve the goal [7]. Rather than focusing on navigation *among* static objects, our work focuses on navigation *of* objects, amidst other dynamic obstacles.

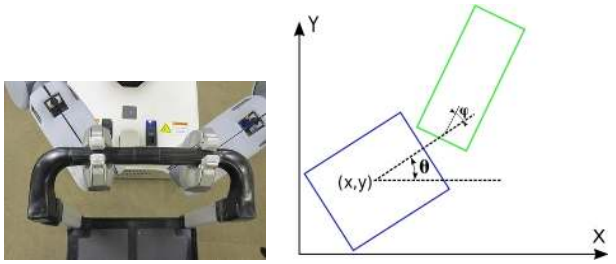
In another mobile manipulation application with a biped humanoid, [9] presented an HRP-2 capable of pushing a human in a wheelchair. As with ASIMO and NAMO, this project described a zero-moment-point (ZMP) offset approach to achieving basic mobility with a mobile object [2]. However, none of these examples with humanoid robots demonstrated robust navigation in dynamic, cluttered environments with tight clearances for the robot, and none took advantage of both arms for articulation. The PR2 mobile manipulation robot used in this study offers a stable, omni-directional base. Its ability to articulate the cart greatly enhances the reachable workspace for the robot since it can now take tight turns in cluttered spaces.

The work that is closest to ours is the approach of Lamiraux, et. al., who demonstrated motion planning for a robot towing a trailer [4]. The trailer is attached to the robot through a single degree of freedom pivot joint. In our approach, we restrict the motion of the cart to be around an (imaginary) pivot point in front of the robot. In contrast to [4], we present a complete solution integrating 3D sensing to develop a system capable of realtime navigation in a cluttered indoor environment.

II. The PR2 Hardware Platform

The hardware used for the experiments in this paper is the PR2 personal robot (Figure 1) which has an omni-directional base and two 7-DOF arms. It is also equipped with a tilting laser scanner mounted to the head, two stereo cameras, an additional laser scanner mounted on the base, and a body-mounted IMU. Encoders on each joint provide joint angle information. The end-effector is a parallel jaw gripper whose fingertips are equipped with capacitive sensor arrays, each consisting of 22 individual cells. The laser scanner mounted on the base is useful both for obstacle detection and localization. The robot's base is approximately 63 cm in both length and width.

The cart used in the experiments in this work is a regular holonomic utility cart. It has casters mounted at all four corners and can thus be pushed in any direction. The top shelf of the cart was removed to reduce the volume of the region in front of the cart that is occluded from the PR2's tilt scanning laser sensor mounted on the head. The PR2 grasps the handle of the cart as shown in Figure 2(a). The grasp is sufficiently rigid to maintain its relative pose with respect to the cart handle.



(a) The PR2 has a firm grasp on the cart handle (right).

(b) State representation: Blue rectangle represents the robot, and θ its orientation. Green rectangle represents the cart, and ψ its orientation in the robot frame.

Fig. 2. Cart grasping (a), and state representation (b)

III. Approach

Navigation with moveable objects is a challenging planning and control problem. Ideally, a planner used for this task should efficiently take advantage of the latent capabilities of the robot, and react quickly to failure given a cluttered and dynamic environment. We make almost no assumptions about the structure of the environment. For simplicity, we assume a known 2D map of the environment as a starting representation. The map was built separately from real sensor data using tools available in the ROS framework [11]. This map is not, however, assumed to be a completely accurate representation of the environment, which can contain both static and dynamic obstacles such as people or other robots. We demonstrate, through experiments, the ability of our approach to deal with such static and dynamic obstacles that are not initially known to the robot. We assume a known geometric and kinematic model for the cart. This helps us in localizing and controlling the cart relative to the robot. Although the task of determining a model for the cart is, by itself, challenging and interesting, it is beyond the scope of this paper and will be examined in greater detail in future work.

Overall our approach consists of a tight integration of three different components: sensing for the cart and the environment, motion planning for the robot and the cart, and control of the robot and the cart along the desired path.

IV. Sensing

Our approach to mobile manipulation builds on components developed for navigation and manipulation with the PR2 [8], [12]. To navigate effectively, our system must be able to differentiate between sensor readings that may correspond to points on the cart or the robot, and those from points in the environment. Sensor readings corresponding to points on the cart are filtered out directly from a 2D costmap representation (Figure 3) of the collision environment if their 2D projection falls within the known polygonal footprint of the cart.

Two approaches were implemented to sense the pose of the cart relative to the robot. First, a checkerboard attached to the cart was used to localize the cart pose relative to the robot

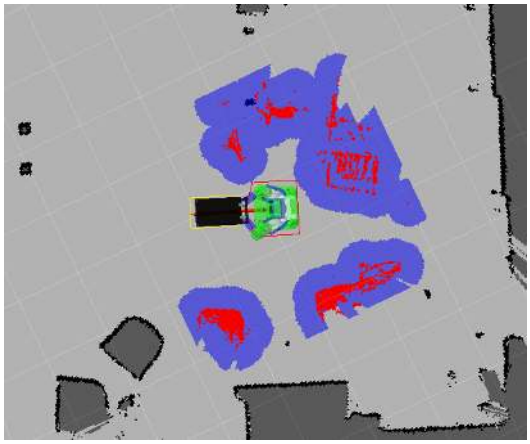


Fig. 3. A view of the local costmap. Red cells are lethal obstacles while blue cells lie within a threshold distance of the nearest obstacle.

using the cameras mounted on the head of the PR2. Second, the known initial positions of the grasps of the two end-effectors on the cart were used to provide a *proprioceptive* estimate of the cart relative pose. Note that this assumes that the cart handle stays rigidly fixed relative to the end-effectors of the two arms, which proved a safe assumption in our experiments. Indeed, we found the proprioceptive estimate to be more stable than the checkerboard estimate due to the visual noise and jitter in the camera.

V. Motion Planning

The motion planner provides global collision-free plans between the start pose (the current pose of the robot) and the final desired position of the robot and the cart. There are several approaches to motion planning for mobile robots that could be applicable in this case, including both graph and sampling-based planners [3]. We choose to use a graph based approach coupled with an anytime planner. As we shall describe in the next few sections, the choice of such a planner allows us to specify candidate motions of the robot and cart system (motion primitives) that allow the robot to navigate tight turns.

We first describe the construction of the graph itself and then we describe the construction of the transitions between the nodes in the graph. We will then briefly describe the algorithm used to search the graph for low-cost solutions.

A. State Representation

To construct the graph, we need to specify the state representation for the nodes of the graph. A full state representation would include the 7 degrees of freedom for each arm, the 3 degrees of freedom of the robot base and the 3 planar degrees of freedom of the cart relative to the robot base. However, as noted earlier, the arms of the robot are constrained by the grasp on the cart, which itself is constrained to planar motion. One simpler representation to eliminate redundancies could be the planar degrees of freedom of both the base of the robot and the cart (in the robot reference frame). Such a representation

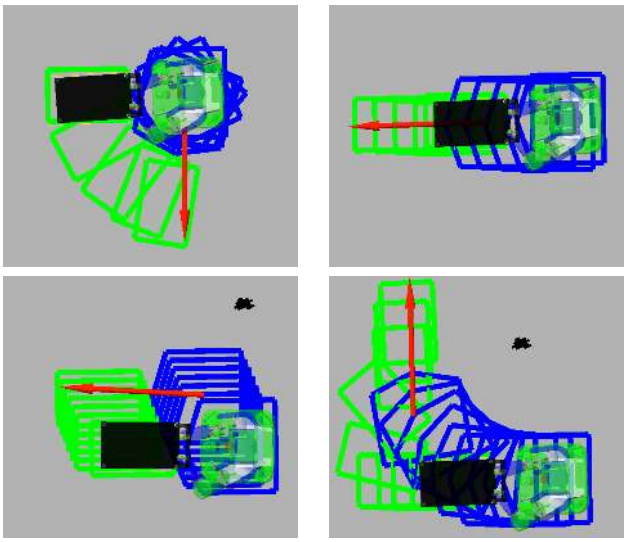


Fig. 4. Four example motion primitives. Top left: rotate in place, Top right: move forward, Bottom left: move diagonally, Bottom right: articulated motion primitive. The red arrow indicates the final pose of the robot at the end of each primitive.

$(x, y, \theta, x_c, y_c, \theta_c)$ would result in a highly controllable 6-dimensional state space for the system. However, we choose instead to restrict the motion of the cart by specifying a fixed point of articulation for the cart in the robot frame. This choice is motivated planning considerations — it reduces the dimensionality of the search space for planning while still retaining enough flexibility to allow the articulation needed to execute tight turns. Our choice of state representation (x, y, θ, ϕ) is shown in Figure 2(b).

B. Transitions

The transitions between nodes in the search graph are defined using a lattice-based planning representation [5], [10]. A lattice-based planning representation discretizes the configuration space into a set of states. The connections between the states are also discretized and every connection represents a feasible path. The lattice representation can be used to specify the motion planning problem as a graph-search. The key advantage of this representation, in contrast to other approaches like 4-connected or 8-connected grids, is that every connection between states is a feasible connection. This makes the lattice-based representation a good choice for constrained systems, such as a robot with an articulable cart.

Since the PR2 robot is omni-directional, we choose to enable transitions that allow the robot to move forwards, along diagonal paths, rotate in place, move backwards and move forwards and backwards while rotating. These transitions are thus *motion primitives* and can be used to generate a search graph of the task space. In order to exploit the controllable degrees of freedom of the robot+cart system, we also designed primitives for simultaneous navigation and articulation. Four of the motion primitives used are illustrated in Figure 4. During the search process, the planner checks the entire motion primitive between parent and child nodes for collisions using the global costmap.

C. Cost Function

In general, the planner can accommodate an arbitrary cost function reflecting such objectives as travel time, risks involved in hard-to-execute maneuvers, distance from obstacles and other metrics. In our experiments, the cost of each edge in the constructed graph corresponded to the time required to execute the motion primitive represented by the edge. The cost of the edge was also increased further when the corresponding motion primitive traveled close to an obstacle. Finally, we also increased the cost of edges for certain motion primitives that were harder to execute.

D. Graph Search

Given a graph defined as above and a cost function associated with each action, an efficient search method is required for finding a solution path. A* search is one of the most popular methods for this problem [1]. It utilizes a heuristic to focus the search towards the most promising areas of the search space. While highly efficient, A* aims to find an optimal path which may not be feasible given time constraints and the dimensionality of the problem. To cope with limited deliberation time, we use an anytime variant of A* — Anytime Repairing A* (ARA*) [6]. This algorithm generates an initial, possibly suboptimal solution quickly and then concentrates on improving this solution while time allows. The algorithm guarantees completeness (for a given graph) and provides bounds on the sub-optimality of the solution at any point of time during the search. Furthermore, this bound, denoted by ϵ, can be controlled by a user. In all of our experiments, we set the initial ϵ to 3.0, implying that the cost of the returned solution can be no worse than 3.0 times the cost of an optimal solution (even though the optimal solution is not known). In nearly all of our experiments, ARA* was able to decrease the bound on on sub-optimality to 1.0 (corresponding to a provably optimal solution) within the time we allocated for planning.

Similar to [5], the heuristics we used were computed online as costs of 2D (x, y) paths that take into account obstacles. These heuristics were computed via a single Dijkstra's search before each planning episode.

E. Local Control

Given a path from the global planner, i.e., a set of waypoints of the form (x, y, θ, ϕ), a *local controller* is responsible for translating this path into commands for the base and arms. It runs at 20 hz in the odometric (rather than the map) frame. It therefore provides robustness against dynamic obstacles and localization errors. It aims to follow the global plan as closely as possible: if the plan is found to be in collision, it will slow down or stop rather than attempt to move around the obstacle. If it is unable to follow the global plan, the local controller aborts, forcing the system to re-plan a global path.

The local controller operates by moving its desired goal along the global plan returned by the planner. Desired base velocity (v_b) and cart (v_c) velocities are determined using a proportional controller (Equation 1).

$$v_b = T_b K_p^b e_b \tag{1}$$

$$v_c = K_p^c e_c \tag{2}$$

where e_b and e_c represent the errors in base pose (expressed in the global frame) and cart pose (expressed in the robot frame), T_b transforms the desired base velocity into the base frame and K_p^b and K_p^c are positive definite gain matrices. The errors are given by,

$$e_b = q_b^d - q_b^a$$

$$e_c = q_c^d - q_c^a$$

The superscripts d and a indicate desired and actual velocities respectively. The desired base and cart command velocities are scaled appropriately based on desired maximum speeds for the base and the cart.

Note that because of our choice to parametrize the base motion by articulating the cart about a fixed point in the robot base frame, the position of the center of the cart and orientation of the cart $q_c = (x_c, y_c, \theta_c)$ are a function of the cart articulation angle ϕ. The position and orientation of the base in the global frame, $q_b = (x, y, \theta)$ is completely specified by the plan. The desired base (v_b) and cart (v_c) velocities are determined using a proportional controller.

The velocities for the base and the cart are forward simulated in the odometric frame to check for collisions against the local costmap. If they result in collision, the velocities are scaled down until a collision is avoided, and failure is declared by the local controller if no scaling is possible. The failure of the controller triggers global re-planning. These steps serve to slow the system down when operating near obstacles, and results in re-planning if failure of the local controller is unavoidable.

F. Cart Articulation

Articulation of the cart is critical for narrow-tolerance navigation problems, such as passing through doors, rounding corners in narrow hallways, or avoiding obstacles in densely cluttered areas. Where previous approaches required error-prone analysis of contact forces and dynamics, the presence of two arms allowed us to produce accurate cart trajectories simply by transforming the desired cart velocity appropriately into desired velocities for the two end-effectors (v_{ee}). These velocities are then mapped into the joint space of each arm using the transpose of the manipulator Jacobian. The net result is a set of desired joint velocities (q_{arm}^d) for each arm that allow the system to track the desired cart velocity:

$$\dot{q}_{arm}^d = J^T v_{ee} \tag{3}$$

VI. EXPERIMENTAL RESULTS

We conducted extensive experimentation to validate the robustness and reliability of our system. The experiments were carried out in a typical office building with furniture and people. The robot was completely autonomous and would follow a sequence of waypoints defined on the building

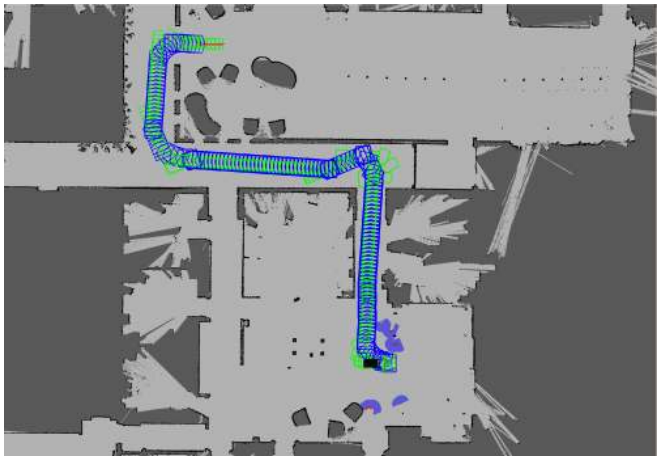


Fig. 5. A long planned path using the lattice-based planner. The red arrow indicates the goal for the robot base, the blue polygon is the footprint of the robot during the plan and the green rectangle represents the cart. Note the frequent use of the articulated motion primitives.

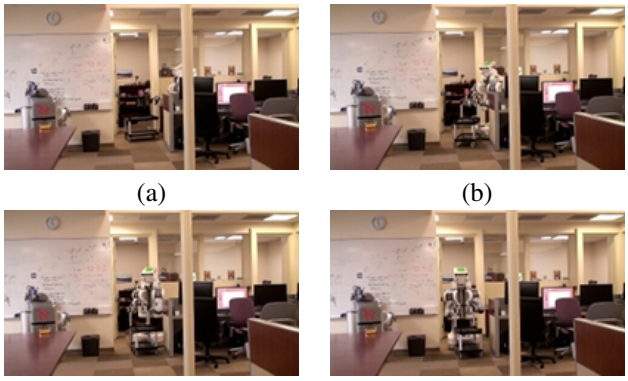


Fig. 6. Articulating the cart to execute a tight turn.

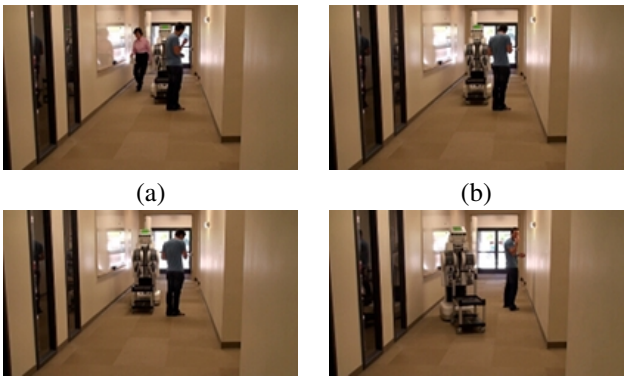


Fig. 7. Stopping in presence of person and then replanning to go around him.

map. Waypoints were chosen such that the robot would have to move through tight spaces and high-traffic areas. The global planner was allocated a maximum planning time of 10 seconds. In most cases, the planner took significantly less time to plan the first solution. Data logged for the global planner included time to first plan, time to final plan, and cost of the plan. An example global plan returned by the planner on the global map is shown in Figure 5 which shows the plan for a distant goal.

The system performed exceptionally well. It was able to handle the presence of people in its vicinity, stopping when they suddenly stepped in front of it, and either re-planning or restarting once they moved away. It never hit any obstacle in the environment. Figure 6 provides a series of snapshots of a run of the robot as it was performing a tight 90 degree turn. Figure 7 contains a series of snapshots showing the robot stopping in time to avoid hitting a person and then planning a path around the person.

Analysis of the data from the planner showed that it was generally successful in finding paths quickly. The analysis is summarized in Table VI for one run lasting nearly an hour in which we repeatedly directed the robot to goals at far regions of the building. The time to first solution corresponds to the time (in seconds) taken to find a solution with the initial value of $\epsilon = 3.0$. The number of expands corresponds to the number of states that were expanded in getting to the initial solution ($\epsilon = 3.0$).

number of planner calls	134
expansions (first solution) (mean)	15157.87
expansions (first solution) (std. dev.)	24529.92
time to first solution (secs) (mean)	1.23
time to first solution (secs) (std. dev.)	1.93
final epsilon (mean)	1.30
final epsilon (std. dev.)	0.61
number of planning failures	25

TABLE I

GLOBAL PLANNER STATISTICS FROM A LONG RUN.

Of the 134 calls that were made to the planner, 25 did not succeed. An examination of these calls showed that they failed much before the allocated final time for the planner expired. There are several reasons that this could have happened. If the robot was blocked by people, no plan would be found for getting out of a tight spot and the planner would fail. Once the people moved out of the way and the costmap was clear, the planner would be able to find a path to its goal. Spurious sensor readings were another frequent cause of failures. Although we attempted to filter out these readings, they were never completely eliminated and would often completely block all plans for the robot. The recovery behavior helped in clearing out these spurious reading but improving the quality of our sensing is essential to achieve longer robust continuous operation.

The addition of the articulated primitives clearly helped the robot traverse tight corridors, made the plans look much more natural, and reduced the total footprint of the robot as it was making turns. Figure 8 illustrates the plans for a turn through -90 degrees. The motion plan on the left was made with a set of articulated motion primitives while the plan on the right was planned without them.

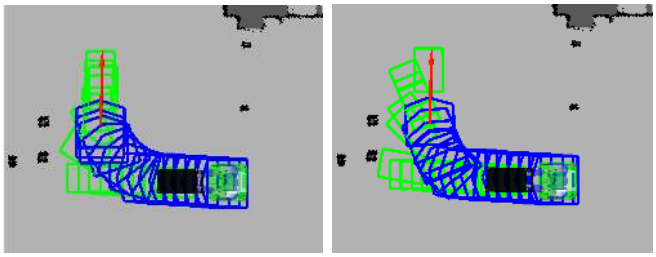


Fig. 8. The overall footprint of the robot when using articulated primitives is much smaller (left) than when it is not using them (right). The red arrow indicates the goal for the robot base, the blue polygon is the footprint of the robot during the plan and the green rectangle represents the cart

VII. CONCLUSIONS AND FUTURE WORK

In this work we have demonstrated a robust and reliable system for cart pushing in a typical office environment. Our approach is, to the best of our knowledge, the first demonstration of such a system in an unconstrained environment. Our method is able to handle tight turns using articulation of the cart, to escape very tight spots requiring the planning of a series of small moves, and can respond to the presence of dynamic obstacles such as humans.

The sensing component of our system was the most frequent source of failure. We plan to address this in future work, possibly through the use of new more accurate 3D sensors or additional sensors mounted on the cart itself. In addition, we plan to incorporate more proprioceptive sensing to detect and account for changes in indoor terrain, e.g. if the cart were to get stuck on a wire on the ground. We also hope to improve the localization of the cart pose relative to the robot by fusing information from proprioception and visual perception.

Our local controller was able to follow paths very closely throughout the experiment, but we would like to be able to track the desired plan at faster speeds. Currently, our speed is limited by safety concerns, by noise in sensing, and by the small amount of compliance in the grasp between the cart and the robot. Safety is a critical issue, especially in the absence of a sensor on the front of the robot. The noise in sensing, especially in the tilting laser sensor, seems to be worsened with faster motion of the robot. The compliance in our grasps implies that if we move too fast, the cart starts overshooting its desired path and could possibly come into contact with the environment.

Our long-term goal is to achieve navigation with arbitrary moveable objects. In this context, our approach is a good starting point since it provides a solution to the motion planning and control of such objects. However, a full real-world solution would also include being able to determine, online, the geometric and physical model for any object. We could attempt to learn the model for the object by exploring different actuation schemes for it online. The learnt model would then form a basis for the design of new feasible motion primitives for the object that the planner could use to plan global plans. Our other goal is to be able to respond faster to changes in the environment and quickly plan around obstacles locally while still following (nominally) the global plan. These issues are currently the focus of active research.

VIII. ACKNOWLEDGMENTS

We would like to thank all members of the PR2 and ROS development teams at Willow Garage.

REFERENCES

[1] 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*, SSC-4(2):100–107, 1968.

[2] S. Kajita, F. Kanehiro, K. Kaneko, K. Fujiwara, K. Harada, K. Yokoi, and H. Hirukawa. Biped walking pattern generation by using preview control of zero-moment point. In *IEEE International Conference on Robotics and Automation*, volume 2, pages 1620–1626. Citeseer, 2003.

[3] Y. Kuwata, Gaston A. Fiore, J. Teo, E. Frazzoli, and How P. J. Motion Planning for Urban Driving using RRT. In *IEEE International Conference on Robotics and Automation*, volume 2, pages 1620–1626. Citeseer, 2003.

[4] F. Lamiraux, D. Bonnafous, and O. Lefebevre. Reactive path deformation for nonholonomic mobile robots. *IEEE Transactions on Robotics*, 6(20):967–977, 2004.

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

[6] M. Likhachev, G. Gordon, and S. Thrun. ARA*: Anytime A* with provable bounds on sub-optimality. In *Advances in Neural Information Processing Systems (NIPS) 16*. Cambridge, MA: MIT Press, 2003.

[7] J. Kuffner M. Stilman. Navigation Among Movable Obstacles: Real-Time Reasoning in Complex Environments. *The International Journal of Humanoid Robotics*, 2(4):479–504, 2005.

[8] E. Marder-Eppstein, E. Berger, T. Foote, B. Gerkey, and K. Konolige. The office marathon: Robust navigation in an indoor office environment. In *Robotics and Automation (ICRA), 2010 IEEE International Conference on*, pages 300–307. IEEE, 2010.

[9] S. Nozawa, T. Maki, M. Kojima, S. Kanzaki, K. Okada, and M. Inaba. Wheelchair Support by a Humanoid Through Integrating Environment Recognition, Whole-body Control and Human-Interface Behind the User. 2008.

[10] M. Pivtoraiko and A. Kelly. Generating near minimal spanning control sets for constrained motion planning in discrete state spaces. In *IEEE International Conference on Intelligent Robots and Systems*, pages 3231–3237, 2005.

[11] M. Quigley, B. Gerkey, K. Conley, J. Faust, T. Foote, J. Leibs, E. Berger, R. Wheeler, and A. Ng. ROS: an open-source Robot Operating System. In *International Conference on Robotics and Automation*, 2009.

[12] Radu Bogdan Rusu, Ioan A. Şucan, Brian Gerkey, Sachin Chitta, Michael Beetz, and Lydia E. Kavraki. Real-time perception guided motion planning for a personal robot. St. Louis, USA, October 2009.

[13] S. Shigemi, Y. Kawaguchi, T. Yoshiike, K. Kawabe, and N. Ogawa. Development of New ASIMO. *HONDA R AND D TECHNICAL REVIEW*, 18(1):38, 2006.

[14] J. Tan and N. Xi. Integrated sensing and control of mobile manipulators. In *IEEE International Conference on Intelligent Robots and Systems*, volume 2, pages 865–870, 2001.

[15] J. Tan and N. Xi. Unified model approach for planning and control of mobile manipulators. In *IEEE International Conference on Robotics and Automation*, volume 3, pages 3145–3152. IEEE; 1999, 2001.

[16] J. Tan, N. Xi, and Y. Wang. Integrated task planning and control for mobile manipulators. *The International Journal of Robotics Research*, 22(5):337, 2003.