

Dynamic Pushing Strategies for Dynamically Stable Mobile Manipulators

Pushkar Kolhe Neil Dantam Mike Stilman

Abstract—This paper presents three effective manipulation strategies for wheeled, dynamically balancing robots with articulated links. By comparing these strategies through analysis, simulation and robot experiments, we show that contact placement and body posture have a significant impact on the robot’s ability to accelerate and displace environment objects. Given object geometry and friction parameters we determine the most effective methods for utilizing wheel torque to perform non-prehensile manipulation.

Index Terms—dynamic stability, mobile manipulation, friction, contact forces

I. INTRODUCTION

In this paper we present, analyze and evaluate three contact and control strategies for dynamically balancing mobile manipulators that push large and heavy objects. Such strategies are useful for transporting hospital stretchers, warehouse crates or a fallen beams that need to be pushed in order to reach a target goal or to create space for the robot to move. [1] Robots designed for mobile manipulation are similar to humans since they can *navigate and manipulate their environments* in order to achieve these tasks. However, in contrast to robots, humans try different postures and contacts. Humans lean and use their entire bodies to perform the task effectively. Typical robots do not consider how their posture or contact will affect their ability to perform the task. [2–5] In this paper we use dynamic analysis and experimental validation to prove that robots can also choose the best configuration for a desired push.

Our work focuses on Sparky, a two-link mobile manipulator that dynamically balances on two wheels. Articulation between the links allows the robot to bend forward and back in order to make contact with environment objects. Prior to contact, an LQR derived PD controller keeps the robot balanced and allows it to navigate. In this paper we choose the contact and analyze the result of applying wheel torque to perform manipulation. We compare moving an object by pushing against a vertical edge, lifting and pushing against a horizontal edge, and colliding with the object. The performance of our techniques is compared via dynamic analysis, dynamic simulation, and experimental results.

This paper is organized as follows. Section II describes the relationship to recent work. Section III explains the robot model and introduces the system dynamics. Section IV presents the distinct strategies used in our experiments. Section V evaluates the strategies in simulation and on the robot, Sparky. Our conclusions are presented in Section VI.

The authors are with the Center for Robotics and Intelligent Machines & School of Interactive Computing at the Georgia Institute of Technology, Atlanta, GA 30332, USA. Emails: pushkar@cc.gatech.edu, ntd@gatech.edu, mstilman@cc.gatech.edu

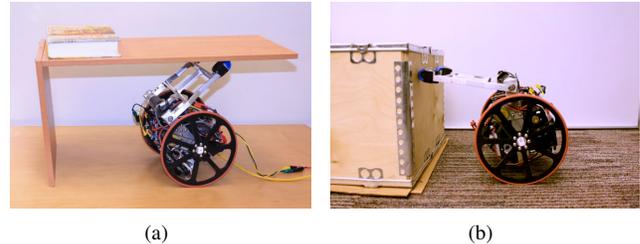


Fig. 1. Robot pushing using some of our strategies. (a) Lift and push is most successful with heavy objects. (b) A simple push strategy.

II. RELATED WORK

Most present day mobile manipulators are robot arms mounted to statically stable mobile bases. [4–7] Such robots have a limited workspace and a limited capacity to control contact and posture. Harada et al. [8, 9] demonstrated that dynamically stable humanoids can lean into objects to manipulate them. Furthermore, Thibodeau et al. [3, 10] used static analysis to show that two-wheeled, dynamically stable bases such as uBot-4 have a greater capacity for resisting external forces without tipping. Experimentally, they found that even a simple balance controller could outperform the pushing capabilities of a statically stable platform. However, we observe that not all pushes are equally effective, and build on previous work by investigating the most effective strategies for dynamic pushing.

Pushing is a common and useful type of manipulation. Lynch and Mason [2, 11, 12] first demonstrated the importance of friction analysis with regard to the types of pushes that could be reliably performed. In fact, [2] presented a pushing planner that could generate an open-loop sequence of controls that reliably manipulates planar objects to a goal. So far, these planners have been based on a simple model of linear contact where the robot approaches the object from a side and pushes it with its body. The work in this paper can be used to extend such planners to cases where the robot uses its internal posture to perform other forms of pushing manipulation.

Harada [13] and Hauser [14] present humanoid pushing and manipulation planners that take into account whole body motion. However, these are largely dedicated to maintaining dynamic stability rather than taking advantage of system dynamics. Yoshida [15] performs another form of whole-body manipulation that pivots an object to change its pressure distribution. To our knowledge, our work is the first to study dynamic pushing from the perspective of choosing contact, posture and wheel torque to maximize the performance of a mobile manipulator when moving heavy objects.

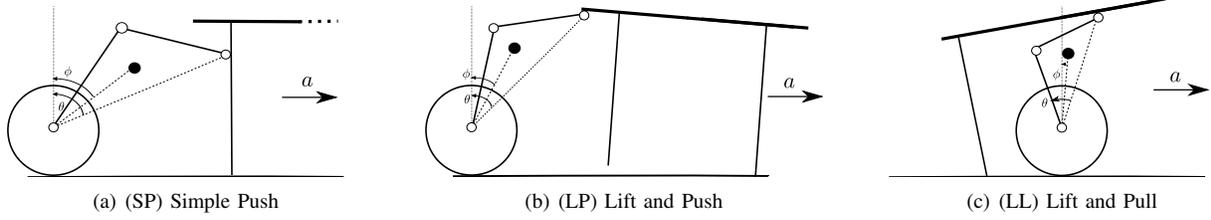


Fig. 2. Some of our pushing strategies. The robot uses different postures to perform the pushing task. Two line segments connect the wheel axle to the robot end-effector and the robot CM respectively. θ gives the angle to the end-effector. ϕ is the angle to the CM.

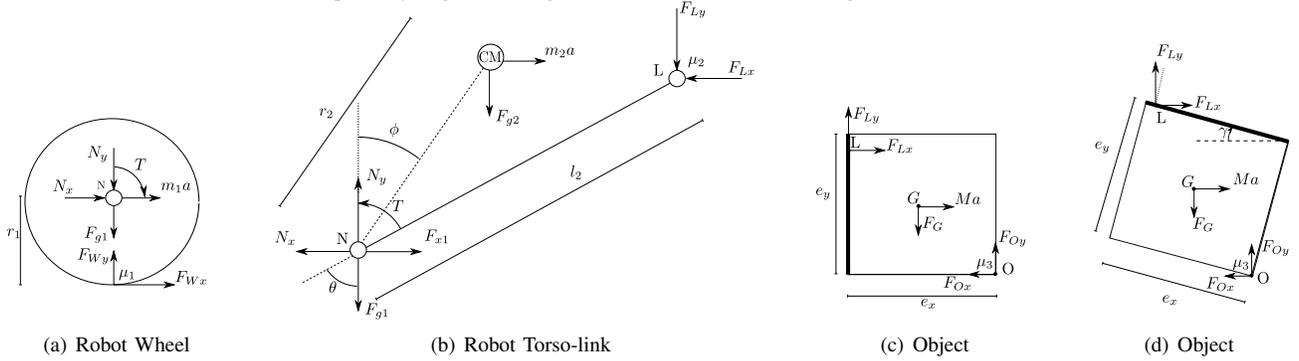


Fig. 3. Free-Body Diagram. (b) represents the robot by a single torso link. (c) refers to an object being pushed using SP as shown in 2(a). (d) presents the object for LP and LL as shown in Fig. 2(b) and 2(c). Thick lines indicate the surfaces of the object that are in contact with the robot.

III. ROBOT MODEL

Sparky is a two-wheeled robot with a torso that consists of a series of links connected by revolute joints. In our dynamic analysis, we model the robot by merging its links into a single torso link that has mass equal to the combined mass of all links. The torso center of mass (CM) is derived from the individual mass centers of the actual links. This paper shows how torque, CM and contact positions affect the dynamics of the robot and the object while pushing and pulling.

When pushing an object, the robot exerts vertical and horizontal forces on the box. This also creates a rotation torque on the box at the point of contact. The sum of moments around the CM of the object, G , must be zero in order to ensure that the object does not rotate and is pushed linearly. To generalize the result, our analysis uses the following parameters: wheel torque, T , end-effector angle, θ and contact angle, ϕ which can be derived for any n -link balancing robot. For various manipulation strategies, these parameters are shown in Fig. 2.

During the Simple Push (SP) strategy, the robot contacts the object at rest and starts pushing. In the Lift-and-Push (LP) and Lift-and-Pull (LL) strategies, the robot initially lifts the object at one end and then applies horizontal force. During LP, the force is aimed towards the object CM, G . During LL, the horizontal force is applied away from G . Initially, the robot uses simple encoder and current feedback to ensure contact and configure itself to execute a particular strategy. Our analysis focuses on identifying the strategy and contact placement that generate the maximum acceleration while maintaining a stable configuration.

In this section, we use free-body diagrams to dynamically model the robot and the object. Section IV builds on this analysis by presenting different contact and action strategies.

Fig. 3 identifies the forces and torques in our three-body system. The system consists of the robot wheel, a merged torso link and an object in two possible configurations.

The force \vec{F}_W is the reaction between wheel and ground. The force \vec{N} is the reaction between the link and wheel. The force \vec{F}_L is the reaction between the box and link. The force \vec{F}_O is the reaction between the ground and box. The linear acceleration of all bodies in the x direction is a . The masses of the wheel, torso link and object are m_1 , m_2 and M , and F_{g1} , F_{g2} and F_G are the consequent weights. The torque T is produced by the motor between the wheel and torso link. Values μ_1 , μ_2 and μ_3 are the coefficients of friction between the wheel and the ground, the link and the object, and the object and the ground. Length r_1 is the wheel radius, l_2 is the length of the torso-link and r_2 is the length of the center of the torso-link from its base. The angle θ is between vertical and the line joining the axle to the end-effector.

For stable pushing it is important that the object being pushed does not rotate. When the object is flat on the ground, as shown in Fig. 3(c), the ZMP of the object should lie within its footprint. When the object is inclined at angle γ , as in Fig. 3(d), it must have zero net moment.

We assume Coulomb friction. The friction force at a sliding contact opposes motion with magnitude $\mu_k f_n$, where f_n is the magnitude of the force normal to the contact plane and μ_k is the kinetic coefficient of friction. While pushing, we need to take care that the link-object and the wheel-ground contacts do not slip. This no-slip condition after contact also ensures that the link has zero angular acceleration.

The linear acceleration of the object is given in Eq. 1.

$$a = \frac{F_{Lx} - \mu_3(F_G - F_{Ly})}{M} \quad (1)$$

When the robot rests its end-effectors on the object and applies force, the object is affected by both the horizontal force component, F_{Lx} , and the vertical force component, F_{Ly} . To find F_{Lx} and F_{Ly} , we derive the equations of motion from Fig. 3(a) and 3(b). Derivations are given in [16].

$$F_{Lx} = -a(2m_1 + m_2) + \frac{T}{r_1} \quad (2)$$

$$F_{Ly} = \frac{T - r_2 \sin \phi F_{g2} - r_2 \cos \phi F_{x2} + l_2 \cos \theta F_{Lx}}{l_2 \sin \theta} \quad (3)$$

Given Eq. 1-3, we find an acceleration function, $a = f(T, \theta, \phi)$, and apply it in the analysis of pushing strategies.

IV. CONTROL STRATEGIES

We present three different types of contact states or parameterized strategies in which the robot pushes an object. Strategies are compared by observing the object's acceleration as given in (1). The different control strategies are Simple Push (SP), Lift and Push/Pull (LP/LL) and Collide.

A. Simple Push (SP)

The SP configuration is shown in Fig. 2(a). The robot leans into the object and then tries to push it. The robot uses its weight and internal joint torques to push the object. The drawback of this strategy is that the maximum force is restricted due to the slip between the end-effector and the object as well as between the wheel and the ground. The constraint is particularly relevant for heavy objects.

SP is valid under the following constraints:

$$|F_{Wx}| \leq \mu_1 F_{Wy} \quad (4)$$

$$|F_{Ly}| \leq \mu_2 F_{Lx} \quad (5)$$

$$\frac{e_x}{2} \geq \left| \frac{1}{F_{Oy}} \left(F_{Ox} \frac{e_y}{2} + F_{Ly} \frac{e_x}{2} + F_{Lx} \frac{e_y}{2} \right) \right| \quad (6)$$

$$F_{Wy} \geq 0 \wedge F_{Lx} \geq 0 \wedge F_{Oy} \geq 0 \quad (7)$$

The inequalities given by (4) and (5) do not let the wheel or the link slip. Inequality (6) is derived from the equation for moment of the object around G and it indicates that the Zero Moment Point of the box must lie within its two contact points to maintain the object's angular acceleration as zero. The inequalities in (7) are required because contact forces cannot be negative.

B. Lift and Push (LP) or Pull (LL)

The LP and LL configurations are demonstrated by Fig. 2(b,c). The robot lifts the object and then tries to push or pull it. Since, the object is lifted, it remains inclined while in contact with the end-effector. The reaction force at the end effector is perpendicular to the object surface. Initially, this strategy involves additional perception and control by the robot to set up a state that lifts the object. Achieving this configuration is not the focus of this paper. LP and LL have slightly different constraints from SP due to to slip at the end-effector and contact forces. Equations (5) - (7) become,

$$|F_{Lx}(\cos \gamma - \mu_2 \sin \gamma)| \leq F_{Ly}(\sin \gamma + \mu_2 \cos \gamma) \quad (8)$$

$$e_y \cos \gamma + e_x \sin \gamma > r_1 + l_2 \cos \theta > e_y \cos \gamma \quad (9)$$

$$\sum M_G = 0 \quad (10)$$

$$F_{Wy} \geq 0 \wedge F_{Ly} \geq 0 \wedge F_{Oy} \geq 0 \quad (11)$$

The inequality given by (8) does not let the link slip. Inequality (9) is a physical constraint on the position of the contact point. Equation (10) is the moment of the object around G and indicates that the box should not rotate. The inequalities in (11) are required because contact forces cannot be negative.

One significant analytical observation is that *LL (Pull) is more effective than LP (Push)*. Consider the simple case of a square object where G is at the center of the square and the robot interacts with it at the top-left corner. The equation of motion for F_{Ly} reduces to (12). [16]

$$F_{Ly} = -F_{Lx} + F_{Oy}(1 + \mu_3) \quad (12)$$

This equation indicates an inverse relationship between F_{Lx} and F_{Ly} . Pushing increases F_{Lx} and decreases F_{Ly} , which in turn decreases the friction between the robot and the object. According to (8), a decrease in F_{Ly} indicates a smaller range of F_{Lx} forces that can be applied without slip. Similarly, decreasing F_{Lx} by pulling, increases F_{Ly} and yields a greater range of forces. Overall, we expect that larger forces can be applied to the object with the LL strategy in contrast to LP without slip. The reader may test this interesting conclusion by lifting the corner of a table on casters and observing that pulling the table is much easier than pushing it while maintaining frictional contact.

For objects that cannot be lifted, the robot can apply a vertical downward force on the object instead of lifting to achieve a similar effect. This strategy requires another minor change to the constraint equations (8) - (10) and a robot capable of applying a downward force using its torso.

C. Collide

In this configuration, the robot generates momentum and transfers it to the object by running into it. This strategy is only used when both of the above strategies fail to generate enough force to overcome static friction. The collision model works on the principle of transfer of momentum. The robot generates linear momentum and strikes the object to overcome static friction. We do not study all the effects of collision here, but we consider it a feasible technique to start the object moving. This effect can be described by the impulse equation Eq. 13.

$$m_{\text{robot}} \Delta v = \int_{t_0}^{t_1} F_{\text{col}} dt \quad (13)$$

V. RESULTS

The simulation and analysis take values for various quantities from the physical robot. Our robot is powered by Maxon DC motors which can give a maximum continuous torque of $1.5 \text{ N} \cdot \text{m}$. The robot's physical parameters are:

$$\begin{aligned} m_1 &= 0.3 \text{ kg}, m_{21} = 3.788 \text{ kg}, m_{22} = 0.220 \text{ kg}, \\ l_{21} &= 0.1300 \text{ m}, l_{22} = 0.1300 \text{ m}, r_1 = 0.0975 \text{ m}, \\ r_{21} &= 0.0254 \text{ m}, r_{22} = 0.08 \text{ m}, \\ \mu_1 &= \mu_2 = \mu_3 = 0.5, \end{aligned}$$

where m_{21} , m_{22} and l_{21} , l_{22} are the masses and the lengths of the first two links of Sparky and r_{21} , r_{22} are the offsets of

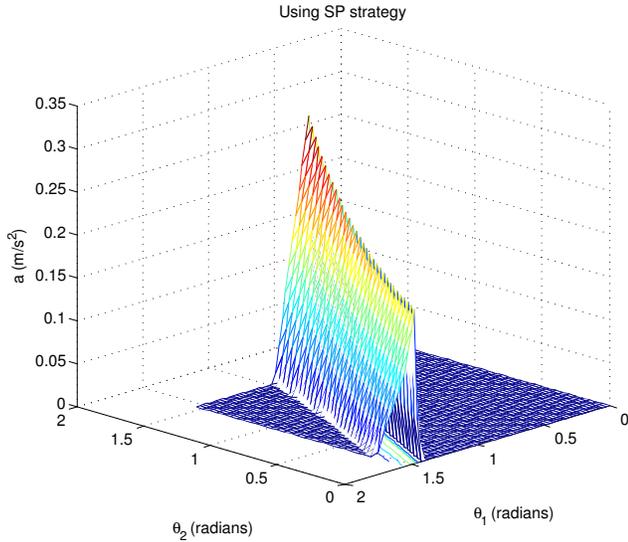


Fig. 4. Variation of acceleration with θ_1 and θ_2 using the simple push strategy for pushing a block weighing 3kg. Peak acceleration of 0.29m/s^2 is achieved at $\theta_1 = 0.725$ rad and $\theta_2 = 1.5$ rad

the positions of the center of mass on these links. θ_1 and θ_2 are the angles between the first and second link of the robot and vertical. The quantities r_2 and ϕ of the torso link are derived quantities which depend on m_{2i} , l_{2i} and c_{2i} where i is the number of links on the robot. For our simulation and analytical experiments we assume that we will be pushing an object like a table having width and height of 15cm and that its center of mass lies at the geometric center as shown in 3(c) and 3(d). The performance of these control strategies is rated by the acceleration of the object that they can achieve. The results for Simple Push and Lift and Pull are shown in Figures 4 and 5. The plots do not take into account collisions that occur for particular combinations of θ_1 , θ_2 and object pose. They represent an object when a torque of $1.5\text{N} \cdot \text{m}$ is applied.

A. Analytical Results

First, consider the plots of acceleration versus θ_1 and θ_2 . Fig. 5 shows negative and positive acceleration. Positive acceleration occurs when the robot lifts and pushes the object. Negative acceleration results from lifting and pulling.

Notice that pulling achieves greater acceleration than pushing. Suppose that the robot is pulling a table after lifting it from underneath. It leverages the mass of the table to counteract the moment that may cause the dynamic system to become unstable while pushing. This results in more normal force at the contact point on the end-effector and the wheels. Hence, more friction can be generated to achieve greater linear force.

The graphs indicate that for effective pulling motion it is better to maintain a stable center of gravity and lift the object with joint torque. This is seen in 5 where we achieve the highest acceleration when the heavier first link is such that $\theta_1 \rightarrow 0$. We can similarly argue for the pushing motion that more acceleration is achieved when we try to lift the table. It can also be shown that greater linear force can be generated

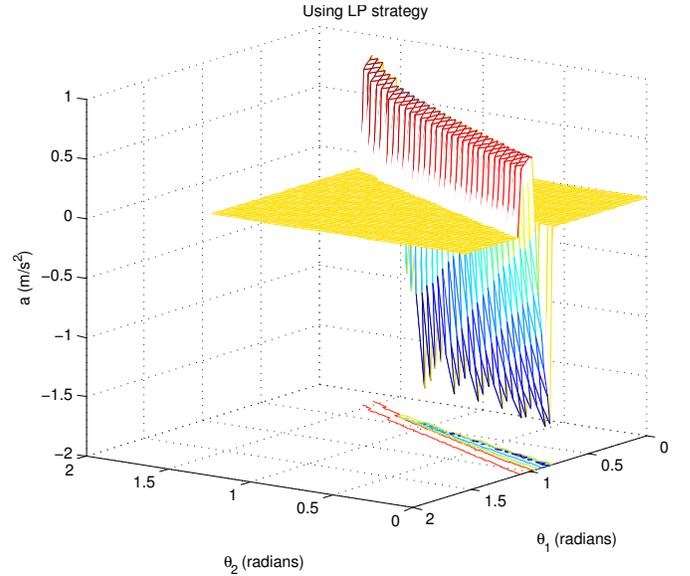


Fig. 5. Variation of acceleration with θ_1 and θ_2 using the Lift and Push/Pull strategy for pushing a block weighing 3kg. Peak acceleration of 0.87m/s^2 is achieved at $\theta_1 = 0.2$ rad and $\theta_2 = 1.5$ rad when pushed. While pulling peak acceleration of 1.6m/s^2 is achieved at $\theta_1 = 0.0$ rad and $\theta_2 = 0.82$ rad.

TABLE I
ANALYTICAL RESULTS OF MAXIMUM MOVABLE MASS

Strategy	ϕ (rad)	θ (rad)	m (kg)	
$\mu_2 = 0.5, \mu_3 = 0.5$	Simple Push	1.03	1.15	3.9
	Lift and Push	0.0	0.0	0
	Lift and Pull	0.17	0.325	8.9
$\mu_2 = 0.5, \mu_3 = 0.3$	Simple Push	1.014	1.15	6.0
	Lift and Push	0.0	0.0	0
	Lift and Pull	0.56	0.25	13.0
$\mu_2 = 0.4, \mu_3 = 0.2$	Simple Push	1.225	1.225	8.4
	Lift and Push	0.121	0.375	12.9
	Lift and Pull	0.065	0.20	17.6

when $\gamma \rightarrow 0$, as it becomes harder to generate enough moment to topple the table. The flat portion of the graph indicates that the end-effector slips at those configurations. Finally, in some cases such as $\theta_1 > 1\text{rad}$ and $\theta_2 > 1\text{rad}$, the end effector does not rise high enough to even touch the table.

On the other hand, Figure 4 shows that the Simple Push strategy succeeds because the robot can lean its own weight into the object. The Simple Push with dynamic balancing is more effective than a statically stable push because of this ability. The vertical force between end-effector and object F_{Ly} helps to reduce the frictional force between object and ground. However it is significantly less than the force observed in Lift and Pull. Beginning from $\theta_1 = 1\text{rad}$, this strategy generates enough force to cause acceleration.

Overall, the graphs demonstrate that the Lift and Pull strategy is more effective than the Simple Push strategy for generating high accelerations.

Table I shows results of the analysis for moving objects with different masses while varying posture for a few differ-

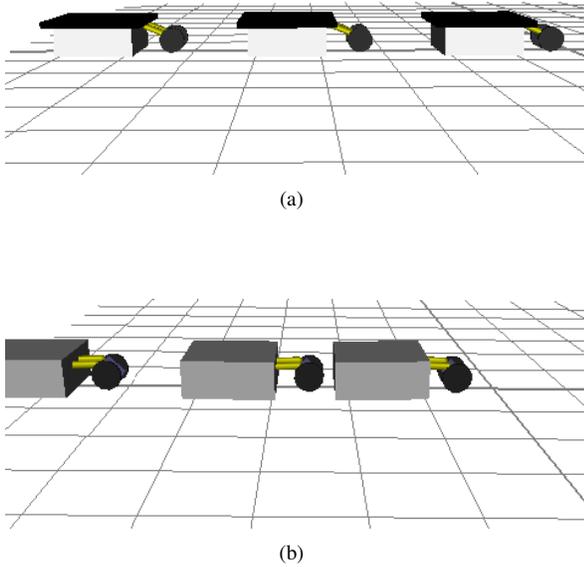


Fig. 6. srLib Simulation for the Lift and Push(LP) and Simple Push (SP) strategy.

ent values of μ_2 and μ_3 . It presents the most massive object that can be moved with a wheel torque of $1.5\text{N} \cdot \text{m}$. Notice that in almost all cases the dynamic strategies allowed the robot to move more load than its own mass of 4.3kg . *The most notable result is that Lift and Pull can move more than twice the mass of Simple Push.*

Lifting an object is analogous to increasing the mass of the robot and decreasing the mass of the object. This means that the robot can apply more torque before the wheels will slip. It is also notable that while Lift and Push will not work when μ_3 is close to μ_2 , or when μ_3 becomes small, Lift and Push can outperform Simple Push in other cases.

B. Simulation Results

In the srLib [17] simulation, we ran experiments to determine the heaviest object that the robot can push. Based on our results, we present two conclusions:

- 1) A dynamically stable robot can move objects heavier than the robot itself while using less torque than robots with static bases.
- 2) The Lift and Pull strategy can move heavier objects than the Simple Push strategy.

We performed several experiments on our robot to verify this analysis. Our main objective was to use posture control to manipulate an object with unknown mass and friction. For this purpose, we designed a control algorithm whose objective is to start the pushing motion and overcome static friction. Based on the results from our analysis, we concluded that the LP strategy works better than the SP strategy. However, for most objects it is difficult for the robot to manipulate them in such a way as to generate the lifting force. Hence, in our control algorithm we prefer the SP strategy. Also, in some cases the height of the object is not sufficiently to allow

TABLE II
REAL WORLD EXPERIMENTS FOR MAXIMUM MOVABLE MASS

Strategy	ϕ (rad)	θ (rad)	m (kg)
Simple Push	0.265	1.22	6

the robot to achieve an end-effector position, θ less than 45 degrees; in such cases, SP is preferable.

C. Experimental Results

In the experimental setup, we pushed a box and added mass to it till it stopped. In our setup for the simple push we pushed a mass of 6kg when we were positioned such that the center of mass was at 0.265 rad and the end effector was at 1.22 rad . This is consistent with the dynamic simulation. Furthermore, since the robot's mass is 4.3kg , it shows that dynamically balancing robots can push objects of mass greater than their own.

VI. APPLICATIONS AND DISCUSSION

Pushing can be used to manipulate an object and change its location. Studies done by Lynch and Mason [2] show how a robot with line contact can use pushing to manipulate an object. The goal of the planner was to avoid obstacles and push an object to a goal position. Their planner had straight pushes and turning pushes, where the control system determined the forces required to make an object turn in a particular radius. Lynch used an algorithm that evaluated discrete pushing configurations based on a cost function. Each line contact was given a cost based on pushing steps, control changes and contact changes to move the object to the goal state. We want to show that posture and internal joint torques are also important parameters in designing pushing planners. Mason and Brost [18] analyze friction cones and the object model to determine the resultant rotation center. Their algorithm can find paths around an obstacle for performing manipulation. However, they do not consider posture and joint torques in their analysis. Our analysis shows that our control strategies can be used to reduce the rotational center while pushing or pulling an object. This effectively means that we can use generic planners used for mobile robots to plan manipulation strategies using pushing.

A. Analysis using Free Body Diagrams

Our robot makes line contact when it rests on an object. Figure 7 shows the free body diagram of a relatively flat object being pushed in the XZ plane by such a robot. α denotes the angular acceleration along Y axis on the body when it is being rotated. The reaction force and the frictional force generated by the placement of the two end points of the end-effector of the robot causes rotational torque.

In the SP case, the robot end-effector pushes against the surface of the box parallel to the YZ plane. This generates a normal force N_x in the X direction. Naturally, this generates a frictional force in the Z direction. In the LP case, the robot end-effector lifts the box using its surface. Since, the surface now is inclined the reaction force extends in the X direction as N_x and also in the Y direction as N_y . Both of these

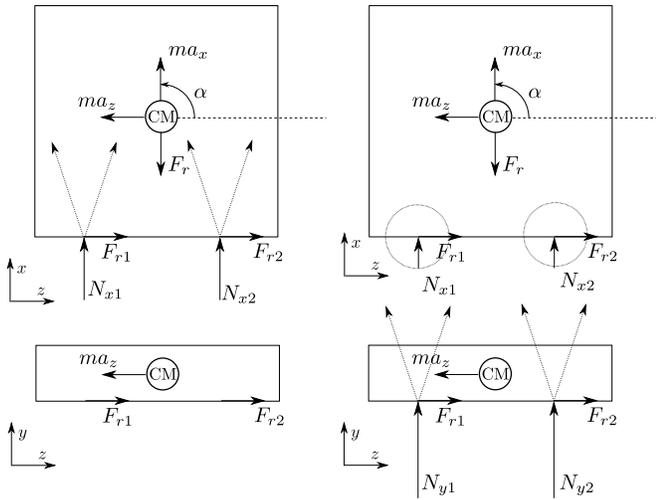


Fig. 7. FBD of a flat object being pushed in the XZ plane. Figure on the left shows a FBD in the SP case, whereas the figure on the right shows a FBD in the LP case. In SP, the cone of friction stretches in the X direction, while in LP case it stretches in the Y direction.



Fig. 8. Sparky pushing a box full of books weighing 6 Kg using the simple push strategy.

normal forces can be used to generate frictional force in the X and Z direction.

Our earlier analysis shows that the LP strategy can be used to generate more forces for pushing or pulling. The LP strategy can also be used to apply rotational torque if the left and right motors of the robot are used to apply different or even opposite torques to the left and right sides of the robot. This will generate different amount of normal forces at the end-effector left and right points creating the moment required to rotate the object. Theoretically, if the object is heavy enough and there is friction between the box and the end-effectors we should be able to generate enough angular torque in the Y direction to rotate the box in its place.

It should be noted that the robot can push into an object to generate a normal in the Y direction for rotating the box without lifting or grasping. This is useful in rotating objects which are smaller than the robot. Our control strategies can be used to manipulate an object in a 2D plane by allowing pushing, pulling and rotation.

VII. CONCLUSION

Dynamic mobile manipulators have more advantages than their static counterparts for performing tasks like pushing and pulling because they can use their posture to exploit the system dynamics. This is most clearly demonstrated by the Lift and Pull strategy, where the friction at the robot

wheel increases due to the transferred weight of the object. In our planner, we demonstrated how lifting or pushing down on an object helps move the object in a circular path. Such manipulation tasks can be performed more elegantly by dynamically balancing mobile manipulators.

Future work will incorporate exteroceptive sensing to fully automate the pushing process. We will also expand our analysis to multiple points of contact. Multiple contacts can create more complex postures for optimizing non-prehensile manipulation tasks like pushing, carrying, and striking.

VIII. ACKNOWLEDGEMENTS

The authors thank Daniel Walker and Tobias Kunz for their help in the development of Sparky. We are grateful to the reviewers for their thorough analysis of our work and insightful suggestions.

REFERENCES

- [1] M. Stilman and J. Kuffner. Navigation among movable obstacles: Real-time reasoning in complex environments. *International Journal of Humanoid Robotics*, 2(1):479–504, December 2005.
- [2] K. M. Lynch and M. T. Mason. Stable pushing: Mechanics, controllability, and planning. *Int. Journal of Robotics Research*, 15(6):533–556, 1996.
- [3] B.J. Thibodeau, P. Deegan, and R. Grupen. Static analysis of contact forces with a mobile manipulator. In *Robotics and Automation, 2006. ICRA 2006. Proceedings 2006 IEEE International Conference on*, pages 4007–4012, 2006.
- [4] Khatib Yokoi, O. Khatib, K. Yokoi, K. Chang, D. Ruspini, R. Holmberg, A. Casal, and A. Baader. Force strategies for cooperative tasks in multiple mobile manipulation systems, 1996.
- [5] O. Ben-Shahar and E. Rivlin. Practical pushing planning for rearrangement tasks. *IEEE Trans. on Robotics and Automation*, 14(4):549–565, 1998.
- [6] R. Holmberg and O. Khatib. Development and control of a holonomic mobile robot for mobile manipulation tasks. *The International Journal of Robotics Research*, 19(11):1066, 2000.
- [7] H. Nguyen, C. Anderson, A. Trevor, A. Jain, Z. Xu, and C.C. Kemp. El-e: An assistive robot that fetches objects from flat surfaces. In *Robotic Helpers, Int. Conf. on Human-Robot Interaction*, 2008.
- [8] K. Harada, S. Kajita, K. Kaneko, and H. Hirukawa. Pushing manipulation by humanoid considering two-kinds of zmps. In *IEEE Int. Conf. on Robotics and Automation*, pages 1627–1632, 2003.
- [9] K. Harada, S. Kajita, F. Kanehiro, K. Fujiwara, K. Kaneko, K. Yokoi, and H. Hirukawa. Real-time planning of humanoid robot's gait for force-controlled manipulation. In *International Conference on Robotics and Automation*, pages 616–622, 2004.
- [10] R. Grupen, B.J. Thibodeau, and P. Deegan. Designing a Self-Stabilizing Robot for Dynamic Mobile Manipulation, 2006.
- [11] M. T. Mason. Compliance and force control for computer controlled manipulators. *IEEE Trans. on Systems, Man, and Cybernetics.*, 11(6), 1981.
- [12] M.T. Mason. *Mechanics of Robotic Manipulation*. MIT Press, 2001.
- [13] K. Harada, S. Kajita, K. Kaneko, and H. Hirukawa. Dynamics and balance of a humanoid robot during manipulation tasks. *Robotics, IEEE Transactions on*, 22(3):568–575, June 2006.
- [14] Victor Ng-Thow-Hing Kris Hauser and Hector Gonzalez-Baos. Multimodal motion planning for a humanoid robot manipulation task.
- [15] E. Yoshida, P. Blazevic, and V. Hugel. Pivoting manipulation of a large object. In *IEEE Int. Conf. on Robotics and Automation*, pages 1052–1057, 2005.
- [16] N. Dantam, P. Kolhe, and M. Stilman. Equations of Motion for Dynamic Mobile Manipulators. In <http://www.golems.org/node/1050>, 2010.
- [17] SNU Robotics Library, Core Dynamics Calculation Library of R-Station. 2008.
- [18] Matthew T. Mason and Randy C. Brost. Automatic grasp planning: an operation space approach. In *ACM '86: Proceedings of 1986 ACM Fall joint computer conference*, pages 124–128, Los Alamitos, CA, USA, 1986. IEEE Computer Society Press.