

Wheelchair Maneuvering with a Single-Spherical-Wheeled Balancing Mobile Manipulator

Cunxi Dai^{1*}, Xiaohan Liu^{1*}, Roberto Shu¹, and Ralph Hollis¹

Abstract—In this work, we present a control framework to effectively maneuver wheelchairs with a dynamically stable mobile manipulator. Wheelchairs are a type of nonholonomic cart system, maneuvering such systems with mobile manipulators (MM) is challenging mostly due to the following reasons: 1) These systems feature nonholonomic constraints and considerably varying inertial parameters that require online identification and adaptation. 2) These systems are widely used in human-centered environments, which demand the MM to operate in potentially crowded spaces while ensuring compliance for safe physical human-robot interaction (pHRI). We propose a control framework that plans whole-body motion based on quasi-static analysis to maneuver heavy nonholonomic carts while maintaining overall compliance. We validated our approach experimentally by maneuvering a wheelchair with a bimanual mobile manipulator, the CMU ballbot. The experiments demonstrate the proposed framework is able to track desired wheelchair velocity with loads varying from 11.8 kg to 79.4 kg at a maximum linear velocity of 0.45 m/s and angular velocity of 0.3 rad/s. Furthermore, we verified that the proposed method can generate human-like motion smoothness of the wheelchair while ensuring safe interactions with the environment.

I. INTRODUCTION

Recent advancements in mobile manipulators have greatly enhanced their utility for assisting humans. A critical aspect is the ability of MMs to operate nonholonomic cart systems, such as shopping carts, luggage carts or hospital beds. These usually have one or two caster wheels and a set of fixed wheels that introduce nonholonomic constraints to their motion. They are essential for transporting cargo or people in environments centered around human activities, including supermarkets, hospitals, and construction sites. Among these, pushing a wheelchair to a specified location is vital for individuals with mobility issues, particularly in places like public transport and healthcare centers. Using robots for wheelchair assistance can significantly reduce a caregiver’s burden and enhance mobility for wheelchair users. When pushing the wheelchair, the robot needs to maneuver the wheelchair with smooth motion while ensuring safe interaction with others surrounding it.

Previous work has been done in maneuvering nonholonomic systems with various platforms. [1]–[3] used bipedal humanoid robots to maneuver heavy objects or carts by computing zero momentum points (ZMP) of interaction with the object and foot placement planning. [4] studied how to change the robot’s posture and body leaning angle based



Fig. 1. Time-lapse picture of the CMU ballbot maneuvering wheelchair around obstacle.

on kinematics to maximize force exertion. While being able to exert large forces via configuration change, bipedal robots face challenges from the inherent discreteness in their locomotion. This can lead to non-smooth movements of the object being pushed if not carefully controlled, undesirable for wheelchair pushing as it may compromise passenger comfort and has reduced energy efficiency on flat surfaces compared to wheeled robots.

Multi-wheeled mobile base robots are also widely used in cart-pushing tasks. [5], [6] studied path planning methods for bimanual mobile-based robots when navigating with a cart, which assumed that the cart can be reoriented by the MM as needed. This does not hold for heavy carts such as wheelchairs occupied by a person. [7]–[9] proposed trajectory-tracking controllers for non-holonomic cart systems that utilize the change of grasping point on the cart to maneuver with a single robot arm. [10] proposed an improved impedance controller for cart-pushing tasks to enhance wheelchair stability and ride quality. Results in these studies indicate that multi-wheeled mobile base robots are capable of producing smoother movements compared to bipedal humanoid robots discussed above. However, a human-height wheeled robot can easily become dynamically unstable if it takes a large impact or accelerates too quickly [11], which led to the adoption of heavy bases with a large footprint in the mentioned studies, hindering mobility in confined areas like hallways such as hallways.

Moreover, previous works lack overall system compliance, which is the compliance in both the arms and the base, either due to hardware limitations or the control methods, key for ensuring safe pHRI in crowded environments.

For wheelchair-pushing tasks, we need to further consider the riding experience of the passenger and the safety of

*Equal contribution.

¹The Robotics Institute, Carnegie Mellon University, Pittsburgh, PA 15213, USA cunxid, xiaohan5, rhollis@cs.cmu.edu, rshumanosalvas@gmail.com.

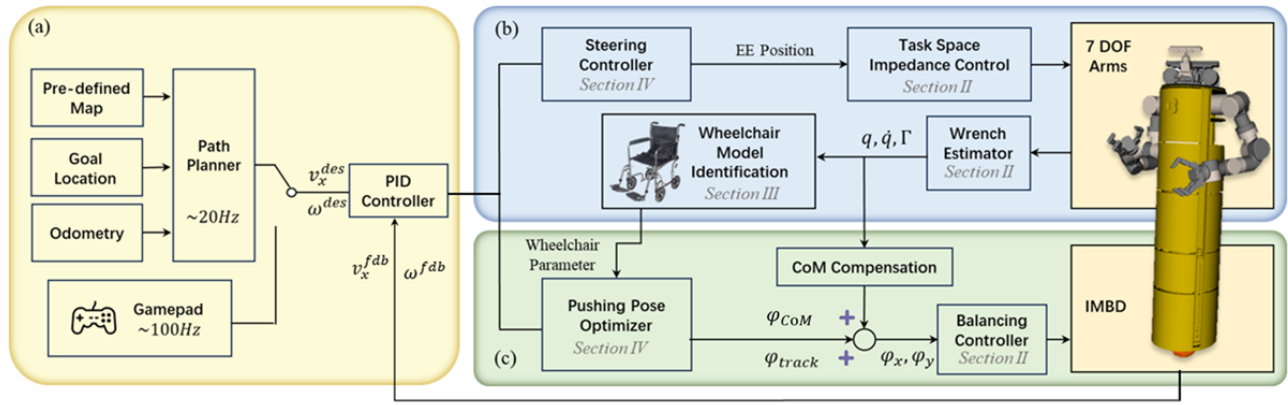


Fig. 2. Control framework diagram. (a) The reference velocity is first generated by the path planner or sent from a gamepad. The velocity is then tracked by the wheelchair-pushing controller by computing the optimal CoM leaning angle and end-effector position commands. (b) The lower-level task-impedance arm controller then tracks the desired end-effector position commands. A wrench estimator is implemented to provide force estimation that is used for wheelchair model identification. (c) The CoM leaning-angle command is tracked with a balancing controller with CoM compensation.

the people around. To address this, our goal is to develop a wheelchair-pushing controller for the CMU ballbot that allows accurate and smooth velocity tracking while maintaining overall system compliance.

The CMU ballbot is a human-sized bimanual robot that operates while balancing on a single spherical wheel, featuring both compliance and omnidirectionality [12]. By leaning its body, the ballbot can change its Center of Mass (CoM) leaning angle to exert large forces such as the required force for human sit-to-stand assistance [13]. The torque-sensing 7-DoF arms provide accurate measurement of the interaction force [14]. These aspects make the CMU ballbot suitable for deployment in caregiving scenarios that require safe pHRI.

In this paper, we present a control framework for the ballbot to dynamically maneuver a wheelchair as shown in Fig. 1, ensuring overall compliance. The controller diagram is shown in Fig. 2. The framework's key elements include a pushing pose optimizer and a steering controller, which together plan a whole-body motion, enabling the ballbot to track the desired wheelchair velocity. To identify the parameters in the wheelchair's model, an extended Kalman filter (EKF) is utilized for online system identification (SI). The main contribution of this paper is a holistic scheme for bimanual dynamic balancing robots to maneuver heavy cart-like systems while ensuring overall compliance.

The remainder of this paper is constructed as follows: Section II introduces the modelling of the CMU ballbot. Section III shows the dynamic model of the wheelchair and online parameter identification. The pushing pose optimizer and steering controller for wheelchair pushing is discussed in Section IV. Finally, we show that the proposed method can accurately track velocity commands with various loads by the experiments described in Section V. Finally, conclusions are summarized in Section VI.

II. BALLBOT MODEL AND CONTROL

This section introduces the ballbot model and implementation of the balancing controller and the arm impedance controller.

A. Kinematics Model

The frame choices are shown in Fig. 3(a), with the origin frame $\{O\}$ centered on the spherical wheel with axes aligned with the inertial frame $\{I\}$, the body frame $\{B\}$ centered at shoulder center with only its z axis aligned with $\{O\}$. The red, green and blue axis in coordinate frames represent x, y and z axis respectively. The arm's task and posture kinematics are defined with respect to body frame $\{B\}$. For both arm, we can write the task relationship between the task coordinates $x \in \mathbb{R}^6$ and the joint configuration coordinates $q \in \mathbb{R}^{10}$ in the following form:

$$x = \mathbf{FK}(q). \quad (1)$$

Here, $q = [q_b, q_a]^T$. $q_b \in \mathbb{R}^3$ is the vector of the body pose defined as $[\phi_x, \phi_y, \phi_z]^T$ w.r.t. $\{O\}$, where ϕ_x, ϕ_y are body leaning angles, and ϕ_z is the body yaw. $q_a \in \mathbb{R}^7$ is the vector of the arm joints. The task coordinate x is defined as $x = [p_x^o, p_y^o, p_z^o, \phi, \theta, \psi]^T$. Where $[p_x^o, p_y^o, p_z^o]^T$ is the position vector w.r.t. $\{O\}$, and $[\phi, \theta, \psi]^T$ are the Z-Y-X Euler angles. The task Jacobian $J(q)$ can then be defined as:

$$J(q) = \frac{\delta \mathbf{FK}(q)}{\delta q} \in \mathbb{R}^{6 \times 10}. \quad (2)$$

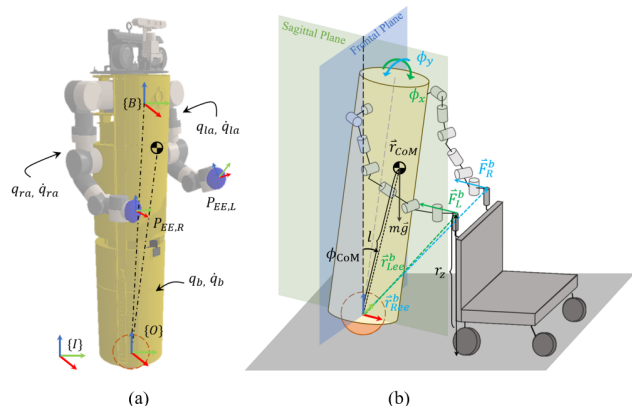


Fig. 3. (a) The ballbot dynamics model. (b) Quasi-static analysis with forces exerted at the end-effectors.

B. Task-space Impedance Controller

Another aspect of care-giving tasks is to ensure user's safety while ensuring safe physical human-robot interaction with others. Towards this goal, a task-space impedance controller is used for the arm's controller to track the desired end effector motion while providing stable physical interaction. The desired impedance behavior between external force F_{ext} and end-effector position error $e_x = x - x_d$ is that of a mass-spring-damper system of the form:

$$\bar{M}_d \ddot{e}_x + \bar{B}_d \dot{e}_x + \bar{K}_d e_x = F_{ext}. \quad (3)$$

The symmetric positive definite matrices \bar{K}_d , \bar{B}_d , and \bar{M}_d are the desired stiffness, damping, and inertia matrix, respectively. F_{ext} is the force exerted at the system. The control law in joint torque space is:

$$\tau^* = J^T \left[\Lambda \ddot{x}_d - \Lambda \bar{M}_d^{-1} (\bar{B}_d \dot{e}_x + \bar{K}_d e_x) + \mu \right], \quad (4)$$

where $\Lambda = (JM^{-1}J^T)^{-1}$, $\mu = \Lambda (JM^{-1}(h - \tau_{fric}) - J\dot{q})$ where $M \in \mathbb{R}^{10 \times 10}$ is the mass/inertia matrix, $h \in \mathbb{R}^{10}$ is the vector of Coriolis, centripetal, and gravity forces, and τ_{fric} is the joint friction.

C. Balancing Controller

The balancing controller is a PID controller cascaded with a PD controller that tracks the desired leaning Center of Mass (CoM) leaning angle ϕ_{CoM} as shown in Fig. 3(b). The two arms each have a mass of 12.9 kg, thus posing considerable CoM change to the robot while moving. A CoM compensator is implemented to maintain the CoM to the equilibrium position using body leaning [14].

III. WHEELCHAIR MODELING

A. Wheelchair Dynamics

The wheelchair has two fixed wheels and two caster wheels in the front, since the caster wheels in the front rotate quickly to the direction of motion, we assume that they have a negligible effect on the system's dynamics [15], [16]. The two rear wheels introduce a nonholonomic constraint such that only motion in the wheelchair's current direction is allowed. A way to model this characteristic is by defining the system state as and $q_w = [x, \theta]$ velocity as $\dot{q}_w = [v_x, \omega]$ w.r.t. the wheelchair frame $\{W\}$, which is the center of the two wheel centers. We can then integrate \dot{q}_w and get

the wheelchair's position w.r.t. the world frame $\{I\}$. Also due to the same constraint, the input wrench exerted at the handle has only 2 effective DoFs, which are the applied force along the x axis of the cart and torque about the z axis, as any lateral force will be dissipated by the constraints [16]. Finally, we assume that the wheelchair is always on the floor. Thus we can faithfully model the wheelchair's dynamics as a secondary planar system. The kinematic energy of the system w.r.t. frame $\{W\}$ can be calculated by:

$$T = \frac{1}{2} m_w v_x^2 - m_w p_x v_x \omega s_\theta - m_w p_y v_x \omega c_\theta + \frac{1}{2} m_w \omega^2 (p_x^2 + p_y^2) + \frac{1}{2} I \omega^2, \quad (5)$$

where s_θ and c_θ are sine and cosine functions of θ . The dynamics equation is given by:

$$\frac{d}{dt} \left(\frac{\partial \mathcal{L}}{\partial \dot{q}_i} \right) - \frac{\partial \mathcal{L}}{\partial q_i} = \Gamma, \quad (6)$$

where Γ is the external wrench applied to the system, including the wrench applied by the robot Γ_w and non-conservative force $N_w(\dot{q}_w)$ such as the viscous force as the wheels rotate. Then, we can reformulate the dynamics equation as:

$$M_w \ddot{q}_w + C_w(\dot{q}_w) \dot{q}_w = \Gamma_w - N_w(\dot{q}_w), \quad (7)$$

where $M_w(q_w)$ is the inertia matrix, and $C_w(\dot{q}_w)$ is the Coriolis term. In the following, the variables describe the properties of the overall wheelchair system (including additional load and passenger). The formulations can be given as:

$$M_w = \begin{bmatrix} m_w & -m_w(p_x s_\theta + p_y c_\theta) \\ -m_w(p_x s_\theta + p_y c_\theta) & I_w + m_w(p_x^2 + p_y^2) \end{bmatrix}, \\ C_w = \begin{bmatrix} 0 & -m_w \omega (p_y s_\theta - p_x c_\theta) \\ 0 & 0 \end{bmatrix}, \quad \Gamma_w = \begin{bmatrix} F^w \\ \tau^w \end{bmatrix}, \quad (8) \\ N_w = \sigma \begin{bmatrix} v_x \\ \omega \end{bmatrix},$$

where σ is a positive viscous coefficient that opposes the system's motion, m_w and I_w are the mass and rotational inertia at the wheelchair CoM respectively. The geometric notations are shown in Fig. 4 (a), where l_w, l_h is the distance between the two rear wheels and the two handles respectively, d is the x -axis offset between the handles and wheels w.r.t $\{W\}$, and p_x, p_y are the distance of CoM w.r.t. $\{W\}$. The input wrenches are typically exerted at the handles as shown in Fig. 4 (b), the mapping between Γ_{input} and Γ_w is :

$$\Gamma_{input} = \begin{bmatrix} \text{Proj}_x \vec{F}_L^w + \text{Proj}_x \vec{F}_R^w \\ \vec{l}_L^w \times \vec{F}_L^w + \vec{l}_R^w \times \vec{F}_R^w \end{bmatrix}. \quad (9)$$

In this context, we assume that the friction between the wheel and the floor is always static friction solely arising from the bearing friction in the rear wheels. We can further model the friction coefficient as:

$$\sigma = [\mu N \quad \mu N l_w / 2]. \quad (10)$$

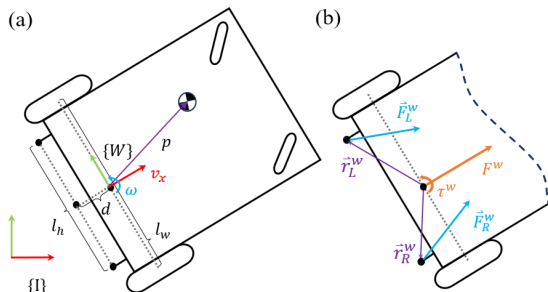


Fig. 4. Planar wheelchair model. (a) Geometric notations. (b) Input force notations.

where μ is the rotational friction coefficient, and N is the force exerted on the wheel. By assuming that the wheelchair's mass m_w is evenly distributed on the four wheels, the force on each wheel is $N = mg/4$, with $g = 9.8 \text{ m}^2/\text{s}$.

B. Wheelchair Model Identification

An EKF is used to estimate the unknown parameters, similar to [16]. We need to estimate the mass m_w , CoM position of the system $[p_x \ p_y]^T$ and the friction coefficient μ . We pick a set of parameters that would be linear in the dynamic equations for ease of computation by selecting $\phi = [m_w \ m_w p_x \ m_w p_y \ I + m_w(p_x^2 + p_y^2) \ \sigma]^T$, these can then be used to calculate the desired parameters. The EKF state vector then consists of:

$$\hat{\mathbf{x}} = [\hat{q}_w^T \ \hat{q}_w^T \ \hat{\phi}^T]^T, \quad (11)$$

And the derivative of the state vector is:

$$\dot{\hat{\mathbf{x}}} = \begin{bmatrix} \dot{\hat{q}}_w \\ \dot{\hat{q}}_w \\ \dot{\hat{\phi}} \end{bmatrix} = \begin{bmatrix} \dot{q}_w \\ \hat{M}_w^{-1} (\Gamma_w - \hat{C}_w \dot{q}_w - \hat{N}_w) \\ 0 \end{bmatrix}. \quad (12)$$

We used an end-effector wrench estimator developed in previous work [17] to estimate the end-effector wrench Γ based on joint torque sensor readings. Then a simple moving average filter is applied to smooth results. The initial values are set to $\phi_0 = [60 \ 0 \ 0 \ 30 \ 0.001]^T$, and they are updated online at 100 Hz. The result of online estimation during experiment is shown in Section V.

IV. WHEELCHAIR PUSHING CONTROLLER

In this part, we propose a pushing pose optimizer based on a quasi-static analysis of the system. The key insight of our controller is to treat the arms as passive spring-like elements and use the body leaning as the dominant factor in velocity control. We also introduce a steering controller that helps find optimal reference positions for end-effectors and minimizes the required leaning angle.

A. Pushing Pose Optimizer

When the ballbot leans, the arms of the robot will be compressed and have force outputs as shown by Eq. (3). If the arm controllers share the same stiffness and damping parameters \bar{K}_d and \bar{B}_d , they will have comparable compression, thus a similar force output. We validated this effect by measuring the EE displacement during a 1-minute run as shown in Fig. 5. The simplification still holds when the length of the arms are not the same. To simplify the control, we assume that the two arms have equal output $\vec{F}_L^b = \vec{F}_R^b = \vec{F}^b$. Since we set the rotational stiffness to be 0, we also assume that there is zero output torque. Then the torque equilibrium at the ball center can be expressed as:

$$-\left(\vec{r}_{Lee}^b + \vec{r}_{Ree}^b\right) \times \vec{F}^b + \vec{r}_{CoM} \times m_{robot} \vec{g} - \tau_{robot} = 0. \quad (13)$$

Here, \vec{r}_{Lee}^b and \vec{r}_{Ree}^b are the position vector of the end-effectors w.r.t. $\{B\}$ as shown in Fig. 3, m_{robot} is the mass of the

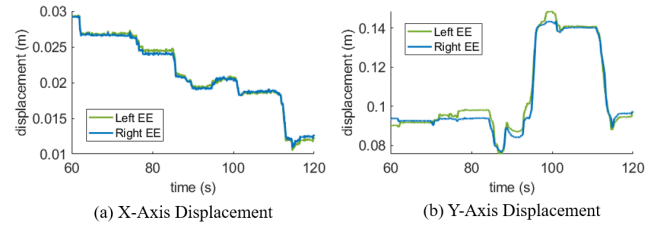


Fig. 5. End-effector displacement during 1-minute wheelchair maneuvering experiment on (a) x axis and (b) y axis w.r.t. $\{B\}$. The maximum displacement mismatch between the two EEs are 0.011 m on the x axis and 0.052 m on the y axis.

ballbot, and τ_{robot} is the net torque result from the friction between the ballbot body and its inverse mouseball drive system (IMBD).

The EEs are commanded to maintain constant height at the handle, so we can assume that no z-axis force will be exerted. Then, for the x and y axes we have:

$$2F_x^b r_z = m_{robot} g l \sin \phi_x, \quad 2F_y^b r_z = m_{robot} g l \sin \phi_y. \quad (14)$$

Here, r_z is the height of the wheelchair handle, and F_x^b and F_y^b are the projection of F^b on the sagittal and frontal plane. Linearizing the equations at $\phi_x, \phi_y = 0$, and we can have a mapping between the output force and body leaning angle:

$$F_x^b = \frac{m_{robot} g l \phi_x}{2r_z}, \quad F_y^b = \frac{m_{robot} g l \phi_y}{2r_z}. \quad (15)$$

Now we move on to the wheelchair system. From Eq. (7), we know that when the wheelchair moves at constant velocity \dot{q}_w , we will have:

$$C_w(\dot{q}_w) \dot{q}_w + N_w(\dot{q}_w) = \Gamma_{input}. \quad (16)$$

Then, expressing the input matrix in the ballbot frame we have:

$$\Gamma_{input} = \begin{bmatrix} 2 \text{Proj}_x R_b^w \vec{F}^b \\ (\vec{r}_L^w + \vec{r}_R^w) \times R_b^w \vec{F}^b \end{bmatrix}. \quad (17)$$

Here, matrix R_b^w is the rotation matrix that transforms a vector in the ballbot frame to the wheelchair frame $\{W\}$.

Based on Eq. (17), we assume that the output force of the end-effectors is influenced by the leaning angle of the body, ϕ_x and ϕ_y . The system is expected to reach the target

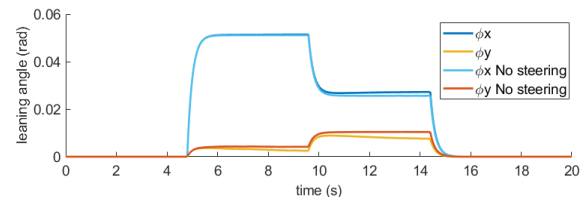


Fig. 6. Comparison of ballbot CoM leaning angle ϕ_x, ϕ_y calculated by the pushing pose optimizer with and without the steering controller. With the steering controller activated, the required CoM leaning angle becomes smaller, indicating less aggressive body movement.

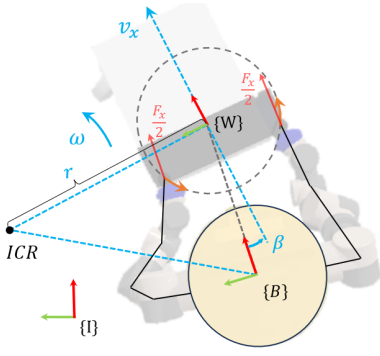


Fig. 7. Wheelchair steering controller schematic. The end-effectors steer the wheelchair around the rotational center of wheelchair, i.e., the midpoint of the two rear fixed wheels, based on the steering angle β .

velocity \dot{q}_w and satisfy the equilibrium equations Eq. (16). Put these equations in matrix form and we have:

$$\underbrace{C_w(\dot{q}_w)\dot{q}_w + N_w(\dot{q}_w)}_f = \underbrace{\begin{bmatrix} 2m_{robot}gl \text{Proj}_x R_b^w \\ m_{robot}gl[r_L^w + r_R^w] \times R_b^w \end{bmatrix}}_A \Phi, \quad (18)$$

where $\Phi = [\phi_x, \phi_y, 0]^T$. $[\vec{a}]_{\times}$ is the cross-product matrix of \vec{a} . For any $\vec{a}, \vec{b} \in \mathbb{R}^3$, we have $[\vec{a}]_{\times} \vec{b} = \vec{a} \times \vec{b}$.

To calculate the optimal body leaning angle, this problem is formulated as an unconstrained quadratic program problem:

$$\min_f (\mathbf{A}\Phi - f)^T \mathbf{Q}(\mathbf{A}\Phi - f) + \Phi^T \mathbf{R}\Phi, \quad (19)$$

where \mathbf{Q}, \mathbf{R} are positive definite weight matrices. The closed-form solution is:

$$\hat{\Phi} = (\mathbf{A}^T \mathbf{Q} \mathbf{A} + \mathbf{R})^{-1} \mathbf{A}^T \mathbf{Q} f. \quad (20)$$

In practice, this is solved online at 100 Hz. At each time step, the controller will update the optimal lean angles. The lean angles are then tracked by the low-level balancing controller.

B. Steering Controller

We want to minimize the ballbot lateral leaning angle ϕ_y to avoid collision between the arms and the body, while keeping the ability to exert torque on the wheelchair for turning. To address this, we propose a wheelchair steering heuristic that plans desirable EE positions for the pushing pose optimizer such that ϕ_y is minimized.

By positioning the ballbot's body such that the center of $\{W\}$ is on the x axis of $\{B\}$, the z-axis torque exerted on the wheelchair can be controlled with ϕ_x and steering angle β as shown in Fig. 7. By steering the wheelchair's direction, we effectively change the z-axis torque exerted on the wheelchair such that the system tracks the desired velocity $[v_x^{des}, \omega^{des}]$ around the Instantaneous Center of Rotation (ICR). The turning radius can be calculated as $r_{ICP} = v_x^{des} / \omega^{des}$, and steering angle β can be calculated as:

$$\beta = \sin^{-1}\left(\frac{\omega l_w}{4v_x^{des} d}\right). \quad (21)$$

The desired end-effector \widehat{r}_{Lee}^b and \widehat{r}_{Ree}^b positions can be calculated as:

$$\begin{aligned} \widehat{r}_{Lee}^b &= [d, 0]^T + R_z(\hat{\beta})^T l_w / 2, \\ \widehat{r}_{Ree}^b &= [d, 0]^T + R_z(\hat{\beta})^T l_w / 2, \end{aligned} \quad (22)$$

where R_z is the rotation around the z axis. From this, the angular velocity can be controlled with steering angle and linear velocity. Due to arm workspace constraint, the steering angle is limited as $\beta \in [-35^\circ, 35^\circ]$.

V. EXPERIMENTS

A. Velocity Tracking with Different Loads

To demonstrate the velocity tracking performance of the proposed planner, we tested with an empty wheelchair (11.8 kg) and a human subject weighing 67.6 kg (79.4 kg in total). The parameters mentioned in Section III are estimated online and updated to the wheelchair model, and the velocity command is sent from a gamepad. The results are shown in Fig. 8, where (a) shows linear and angular velocity tracking with different loads, and (b) shows the online estimated mass. When maneuvering an empty cart, the average system response time is 1.6 s for a 0.2 m/s step in linear velocity, and 1.7 s for a 0.15 rad/s step in angular velocity. When maneuvering a wheelchair with a person, the average system response time is 1.1 s for a 0.2 m/s step in linear velocity, and 1.8 s for a 0.18 rad/s step in angular velocity. The online estimation converges in 10.2 s with a 3.2 kg (4%) steady-state error when pushing the wheelchair with a human, and converges in 13.4 s with a 1.1 kg (9.3%) when pushing an empty wheelchair. The maximum velocity achieved in our experiments with a load on a wheelchair (34.6 kg in total) is 0.45 m/s of linear velocity and 0.3 rad/s of angular velocity. Due to the limited space in the room, we did not conduct further tests on the system's maximum speed. Furthermore, the system demonstrated the ability to effectively turn the wheelchair in place, which is important to navigating in narrow spaces such as hallways and elevators.

B. Motion Smoothness

In this experiment, we compared the wheelchair's acceleration when maneuvered by the ballbot against when maneuvered by a human as a metric to evaluate motion smoothness [18]. Initially, we remotely operated the ballbot to navigate through the lab around an obstacle as shown in Fig. 1, followed by a human attempting to replicate the same path. An Intel Realsense T265 sensor is mounted on the wheelchair to record the trajectory and acceleration data as shown in Fig. 9 (b). We assessed the wheelchair's motion smoothness using the acceleration norm as a metric, with a lower acceleration norm suggesting gentler accelerations or decelerations, thereby indicating smoother movements.

The data showed that the ballbot's maneuvering resulted in 25.3% higher acceleration norm on the x axis, 21.7% on the y axis, and 19% on the z axis w.r.t. $\{B\}$ compared to human maneuvering. This result showed that our approach can

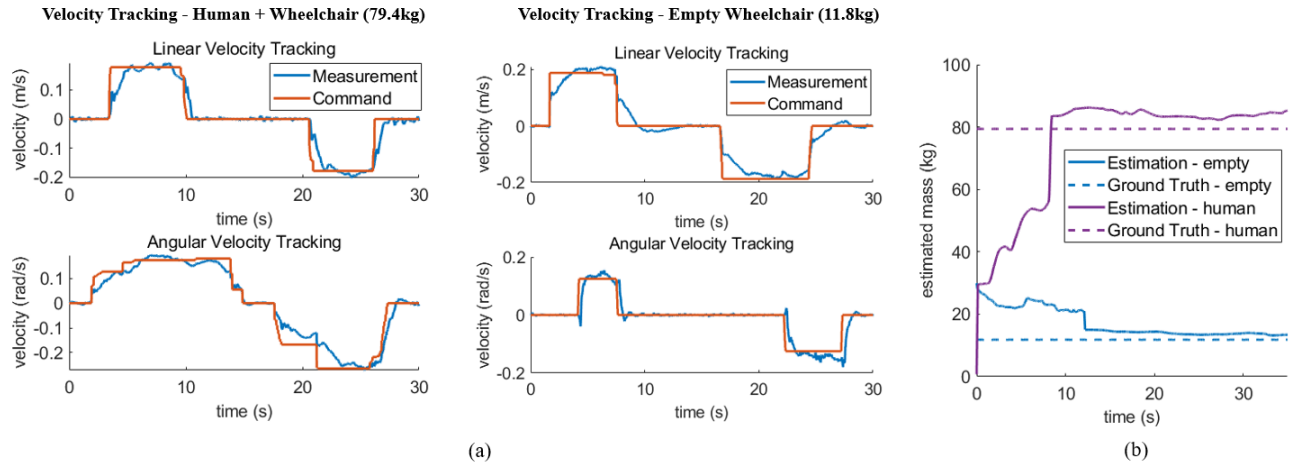


Fig. 8. (a) Velocity tracking performance. The left figure shows results when a human sits in the wheelchair and the right figure shows results with an empty wheelchair. (b) Online mass estimation with human (79.4 kg in total) and empty cart (11.8 kg).

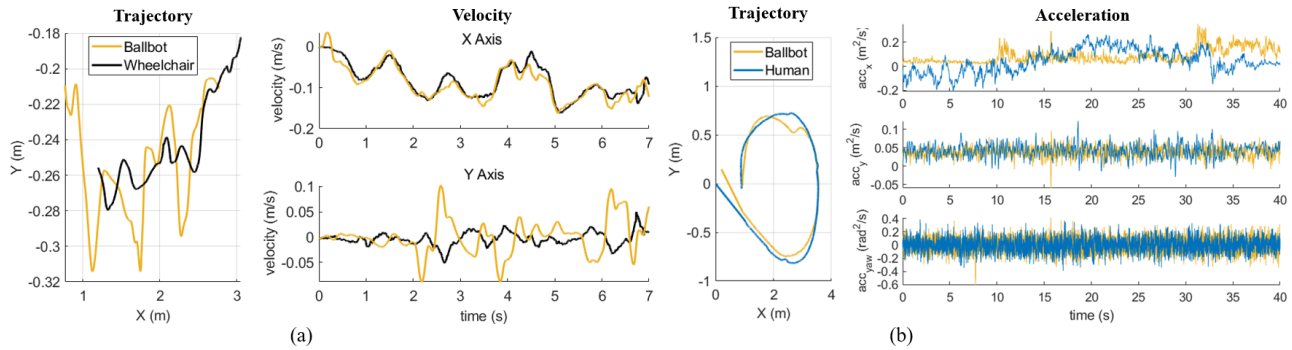


Fig. 9. (a) The trajectory (left figure) and velocity (right figure) of both the ballbot and the wheelchair when disturbed while pushing. (b) The trajectory (left figure) and acceleration (right figure) of the wheelchair when pushed with the ballbot and human.

generate human-like motion smoothness of the wheelchair, which is essential for ensuring passenger’s comfort.

C. Physical Compliance

Another important goal of the proposed controller is the ability to ensure safe pHRI with overall compliance. To validate this, we manually disturbed the ballbot’s body while it is tracking a linear velocity command of 0.1 m/s. The velocity and position of both the ballbot and wheelchair are measured to analyze how the impact is transmitted to the wheelchair. As suggested in Fig. 9 (a), the x-axis velocity of the wheelchair closely mirrors that of the ballbot. On the y axis, the disturbance is more evidently weakened due to overall compliance, resulting in a less aggressive velocity change of the wheelchair compared to the ballbot. Furthermore, we tested the overall compliance of the system by measuring the minimum force required to move the wheelchair by pushing on ballbot. The measured required force is 22.7 N on the x axis and 12.3 N on the y axis w.r.t. $\{B\}$ for the system to move, showing that the overall compliance of the system can ensure safety for people that might collide with the system, and can be manually stopped with small amount of force.

D. Navigating with the Wheelchair

A vital task for wheelchair maneuvering is to navigate it to the desired position while avoiding obstacles. In this experiment, we integrated the proposed controller with the ROS navigation framework [19]. The task is to navigate the wheelchair with a 34.6 kg load across the lab in a known map and avoid predefined obstacles as shown in Fig. 10 (a). The planner replans at 20 Hz frequency online and sends real-time velocity commands to the controller. Acceleration constraints were added to the planner to generate feasible commands for the wheelchair-pushing system, avoiding drastic movements. Fig. 10 (c) shows the time-lapse picture of this experiment. The system successfully reached the given goal location without collision. This experiment shows the controller’s ability to effectively track real-time velocity commands and compatibility with the existing navigation pipeline.

VI. CONCLUSION

We present a control framework that enables the CMU ballbot to maneuver a wheelchair while maintaining balance. The core idea is to utilize whole-body motion to maneuver while maintaining overall compliance. The proposed method is evaluated in real hardware experiments, showing that the approach can perform desirable maneuvers while ensuring



Fig. 10. Navigation across lab with predefined obstacles. (a) Map with pre-defined obstacles. The markers indicate the start position (left marker) and goal position (right marker), and the yellow line shows the actual trajectory of the ballbot. (b) Time-lapse picture of the experiment. Yellow arrow indicates ballbot's velocity direction.

smooth wheelchair motion and safe pHRI. In future work, autonomous affordance detection will allow grasping point identification. It will also be necessary to evaluate the ballbot pushing a wheelchair up and down ADA-compliant ramps [20]. Furthermore, it is likely that the proposed framework can be applied to similar non-holonomic cart systems such as wheelbarrows and shopping carts.

ACKNOWLEDGMENT

This work was supported in part by NSF grant CNS-1629757.

REFERENCES

- [1] S. Nozawa, Y. Kakiuchi, K. Okada, and M. Inaba, "Controlling the planar motion of a heavy object by pushing with a humanoid robot using dual-arm force control," in *2012 IEEE International Conference on Robotics and Automation*. St Paul, MN, USA: IEEE, May 2012, pp. 1428–1435.
- [2] J. C. Vaz and P. Oh, "Material handling by humanoid robot while pushing carts using a walking pattern based on capture point," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*, May 2020, pp. 9796–9801.
- [3] L. Hawley and W. Suleiman, "Control strategy and implementation for a humanoid robot pushing a heavy load on a rolling cart," in *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Sep. 2017, pp. 4997–5002.
- [4] M. P. Polverini, A. Laurenzi, E. M. Hoffman, F. Ruscelli, and N. G. Tsagarakis, "Multi-contact heavy object pushing with a centaur-type humanoid robot: Planning and control for a real demonstrator," *IEEE Robotics and Automation Letters*, vol. 5, no. 2, pp. 859–866, Apr. 2020.
- [5] J. Scholz, S. Chitta, B. Marthi, and M. Likhachev, "Cart pushing with a mobile manipulation system: Towards navigation with moveable objects," in *2011 IEEE International Conference on Robotics and Automation*, 2011, pp. 6115–6120.
- [6] M. Schulze, F. Graaf, L. Steffen, A. Roennau, and R. Dillmann, "A trajectory planner for mobile robots steering non-holonomic wheelchairs in dynamic environments," in *2023 IEEE International Conference on Robotics and Automation (ICRA)*. London, United Kingdom: IEEE, May 2023, pp. 3642–3648.
- [7] S. Aguilera and S. Hutchinson, "Control of cart-like nonholonomic systems using a mobile manipulator," in *2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. Detroit, MI, USA: IEEE, Oct. 2023, pp. 6801–6808.
- [8] N. Methil and R. Mukherjee, "Pushing and steering wheelchairs using a holonomic mobile robot with a single arm," in *2006 IEEE/RSJ International Conference on Intelligent Robots and Systems*. Beijing, China: IEEE, Oct. 2006, pp. 5781–5785.
- [9] N. S. Methil and R. Mukherjee, "Adaptive control of a wheelchair-pushing holonomic robot subject to input constraints," in *Unmanned Systems Technology X*, vol. 6962. SPIE, Apr. 2008, pp. 629–640.
- [10] Y. Fujimoto and T. Murakami, "An improvement method of compliance control in pushing operation by mobile manipulator," in *IECON 2010 - 36th Annual Conference on IEEE Industrial Electronics Society*. Glendale, AZ, USA: IEEE, Nov. 2010, pp. 1447–1452.
- [11] T. Lauwers, G. Kantor, and R. Hollis, "A dynamically stable single-wheeled mobile robot with inverse mouse-ball drive," in *Proceedings 2006 IEEE International Conference on Robotics and Automation, 2006. ICRA 2006.*, 2006, pp. 2884–2889.
- [12] R. Shu and R. Hollis, "Momentum based whole-body optimal planning for a single-spherical-wheeled balancing mobile manipulator," in *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. Prague, Czech Republic: IEEE, Sep. 2021, pp. 3221–3226.
- [13] M. Shomin, J. Forlizzi, and R. Hollis, "Sit-to-stand assistance with a balancing mobile robot," in *2015 IEEE International Conference on Robotics and Automation (ICRA)*, May 2015, pp. 3795–3800.
- [14] R. Shu and R. Hollis, "Development of a humanoid dual arm system for a single spherical wheeled balancing mobile robot," in *2019 IEEE-RAS 19th International Conference on Humanoid Robots (Humanoids)*. IEEE, pp. 499–504. [Online]. Available: <https://ieeexplore.ieee.org/document/9034999/>
- [15] G. Campion and W. Chung, "Wheeled rob 17. wheeled robots," *Part B*.
- [16] S. Aguilera, M. A. Murtaza, J. Rogers, and S. Hutchinson, "Modeling and inertial parameter estimation of cart-like nonholonomic systems using a mobile manipulator," in *2023 IEEE International Conference on Robotics and Automation (ICRA)*. London, United Kingdom: IEEE, May 2023, pp. 3073–3079.
- [17] R. Shu, "An agile and dexterous balancing mobile manipulator robot."
- [18] M. Schulze, F. Graaf, L. Steffen, A. Roennau, and R. Dillmann, "A trajectory planner for mobile robots steering non-holonomic wheelchairs in dynamic environments," in *2023 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, pp. 3642–3648. [Online]. Available: <https://ieeexplore.ieee.org/document/10161082/>
- [19] E. Marder-Eppstein, E. Berger, T. Foote, B. Gerkey, and K. Konolige, "The office marathon: Robust navigation in an indoor office environment," in *2010 IEEE International Conference on Robotics and Automation*, 2010, pp. 300–307.
- [20] B. Vaidya, M. Shomin, R. Hollis, and G. Kantor, "Operation of the ballbot on slopes and with center-of-mass offsets," in *2015 IEEE International Conference on Robotics and Automation (ICRA)*, 2015, pp. 2383–2388.