Abstract—With the growth of elderly population in our society, technology will play an important role in providing functional mobility to humans. From the perspective of human safety, it is desirable that controllers for walk-assist robots be dissipative, i.e., the energy is supplied by the human to the walker, while the controller modulates this energy, also the motion of the walker, while dissipating this energy. The simplest form of a dissipating controller is a brake, where resistive torques are applied to the wheels proportional to their speeds. The fundamental question that we ask in this paper is how to modulate these proportionality gains over time for the two wheels so that the walker can perform point-to-point motions in the static space. The unique contribution of this paper is a novel way in which the theory of differential flatness is used to plan the trajectory of these braking gains. Since the user input force is not known prior, the theory of model predictive control is used to periodically compute the trajectory of these braking gains. The simulation results show that the walking assist robot, along with the structure of this proposed control scheme, can guide the user to a goal accurately.

I. INTRODUCTION

As the elderly population grows rapidly, walking assist robots will continue to be an important research topic in our society [1-3]. This research aims to provide assistance to the elderly during walking and make a difference to their overall health. To enhance their safety and convenience, intelligent walking aids are needed. Robotics contains several useful technology components for motion control, sensing, and computational intelligence. It is timely to design robot walking helpers for the elderly by integrating these technologies.

Many robot walking helpers have been proposed to assist human walking [1-11]. Systems that support useful functions such as guidance [1,5,7], obstacle avoidance [8,9], and health monitoring [1,7] have been developed. In general, these robot walking helpers can be classified into two types: (i) active and (ii) passive. The active robot walking helpers [1-5] use servo motors to provide guidance to the user, while actively adding energy to the system. The passive walking helpers [8-11] move only by user-applied forces and controlled brakes are used to steer thewalker while constantly extracting energy out of the system. Passive walkers use dissipative control laws. With this property of dissipating energy, they are inherently safe and avoid energy build up. To effectively use a passive robot walking helper, it is important to appropriately control the braking torque in accordance with user applied force so that it can be steered to the goal.

To control the robot walking helper, Spenko et al. [7] used a variable damping model to increase walking stability. Hirata et al. [8] proposed an adaptive motion control algorithm for obstacle/step avoidance and gravity compensation. Chuy et al. [9] used the passive behavior to enhance the interaction between the user and support system. They controlled an active system in passive mode. Ryu and Agrawal [10] proposed a two-phase passive control algorithm for guiding a user to attain desired position and orientation, while allowing for small errors. Efficient trajectory planning and control of a passive robot still remains an open question.

In this paper, we adopt braking control law on the wheels to differentially steer a walking helper. This control law is both passive and dissipative. An open question is how to choose the braking gains so that the vehicle achieves a desired final position and orientation. The novelty of this paper is to use the method of differential flatness to select time varying trajectories of braking gains so that the vehicle achieves a desired final position and orientation [14, 13]. In this approach, the nonlinear structure of the vehicle dynamic equations under the braking control law is explored to seek the property of differential flatness. In this model, the braking coefficients are treated as the new control inputs for the system.

Once this property of differential flatness is demonstrated for the system under the braking control law, the states and inputs are represented as functions of flat outputs and their derivatives. The flat outputs are then parameterized by a set of mode functions that fit the boundary conditions. A feasible trajectory for the proportional gains, generated by the passive control law, is found by nonlinear programming. Since the applied forces are not known prior, model predictive control (MPC) is used to find the solution repetitively. Finally, simulations of this passive robot walker are performed to demonstrate the effectiveness of the proposed approach.

The remainder of this paper is organized as follows: Section II describes the dynamic equations of the passive robot walking helper. In Section III, dynamic feedback linearity is addressed. Trajectory planning and model predictive control are described in Section IV and V. Section
II. THE ROBOT WALKING HELPER AND ITS MODEL

The robot walking helper [11] is shown in Fig. 1. It contains the support frame, two wheels with servo brakes, encoders, two passive casters, an ultrasonic sensor array, a force sensor, and a controller. The support frame has a semi-enclosed design that allows the user to walk at the center of the helper and provides good support and stability. In order to obtain high brake torque, a gear system with a 2.5 magnification is used between the servo brake and the rear wheel. The encoders are used to measure the wheel speed, while force sensors detect the user applied force. The servos provide proper braking force for controlling the motion of the robot walking helper. To design the controller, its dynamics is described next.

Fig. 2 shows the robot walking helper configuration in Cartesian coordinates [10,12] given by

$$ q = [x, y, \theta] $$

where \( x, y \) are the coordinates of the center of mass and \( \theta \) the heading angle of the robot. With the assumption of no-slip condition at the wheel contact points, the velocity of the wheel centers are parallel to the heading direction. Hence, \( \dot{q} \) can be expressed as

$$ \dot{q} = S(q) \dot{W} $$

where \( \dot{W} \) is the vector of the heading speed \( v \) and turning speed \( \omega \), i.e. \( \dot{W} = [v, \omega]^T \). \( S(q) \) is given by

$$ S(q) = \begin{bmatrix} \cos \theta & 0 \\ \sin \theta & 0 \\ 0 & 1 \end{bmatrix} $$

The equations of motion with the user-applied force \( F \) and the no-slip constraint force are written as

$$ M \ddot{q} = E \tau + DF - C \lambda $$

where

$$ M = \begin{bmatrix} m & 0 & 0 \\ 0 & m & 0 \\ 0 & 0 & I \end{bmatrix}, \quad E = \begin{bmatrix} \cos \theta/r & \cos \theta/r \\ \sin \theta/r & \sin \theta/r \\ b/r & -b/r \end{bmatrix}, \quad \tau = \begin{bmatrix} \tau_r \\ \tau_l \end{bmatrix}, \quad D = \begin{bmatrix} \cos \theta \\ \sin \theta \\ 0 \end{bmatrix}, \quad C = \begin{bmatrix} \sin \theta \\ -\cos \theta \\ 0 \end{bmatrix} $$

Here, \( m \) is the robot mass, \( I \) the moment of inertia of the robot, \( r \) the wheel radius, \( b \) half distance between the two wheels, \( \tau_r \) and \( \tau_l \) the motor torques on the right and left wheels, and \( \lambda \) the constraint force.

By differentiating Eq. (2), one gets \( \ddot{q} = S \dot{W} + S \dot{\dot{W}} \). Substituting this \( \ddot{q} \) into Eq. (4), pre-multiplying by \( S^T \), the constraint force \( \lambda \) is eliminated and a reduced model is obtained as

$$ \begin{bmatrix} \ddot{q} \\ \dot{V} \end{bmatrix} = \begin{bmatrix} S \dot{W} \\ 0 \end{bmatrix} + \begin{bmatrix} 0 \\ (S^T MS)^{-1} S^T E \end{bmatrix} \tau + \begin{bmatrix} 0 \\ (S^T MS)^{-1} S^T D \end{bmatrix} F $$

To make the robot passive and dissipative, the control law is chosen as

$$ \tau_r = -K \dot{\theta}_r, \quad \tau_l = -K \dot{\theta}_l $$

where \( \dot{\theta}_r \) and \( \dot{\theta}_l \) are the angular speeds of the right and the left wheels, respectively. \( K_r \) and \( K_l \) are non-negative parameters. With the no-slip condition, the angular speeds \( \dot{\theta}_r, \dot{\theta}_l \) can be calculated as

$$ \dot{\theta}_r = (v + b \omega) / r, \quad \dot{\theta}_l = (v - b \omega) / r $$

Substituting Eqs. (7) and (8) into Eq. (6), we can obtain the dynamic equations of the passive robot given by

$$ \begin{bmatrix} \ddot{q} \\ \dot{V} \end{bmatrix} = S \dot{W} + AK + BF $$

where

$$ A = \begin{bmatrix} -\frac{v + b \omega}{mr^2} & -\frac{v - b \omega}{b(v + b \omega)} \\ \frac{b(v + b \omega)}{Ir^2} & \frac{b(v - b \omega)}{b(v - b \omega)} \end{bmatrix}, \quad K = \begin{bmatrix} K_r \\ K_l \end{bmatrix}, \quad B = \begin{bmatrix} 1/m \\ 0 \end{bmatrix} $$

The non-negative parameters in \( K \) can now be regarded as the new control inputs which need to be planned to steer the vehicle from its current state to the desired goal state. To
design the feedback controller, we further transform the dynamic equations to be static and/or dynamic feedback linearizable by using the differential flatness approach, described in the next section.

### III. Dynamic Feedback Linearization

In the differential flatness approach [13,14], we first select suitable flat outputs and then express all state variables and input in terms of the flat outputs and their derivatives. Here, we find that the passive robot system is differentially flat with the flat outputs \((y_1, y_2) = (\dot{\theta}, \dot{\theta})\). However, the system model in Eq. (9) is not statically feedback linearizable. Thus, we first introduce input transformation with

\[
\begin{bmatrix}
X_1 \\
X_2
\end{bmatrix} = AK + BF
\]  

(11)

The dynamic system can be demonstrated to be static feedback linearizable through one prolongation of \(X_1\) by considering an additional state defined as \(X_1 = \dot{X}_1\). The extended system is given by

\[
\begin{align*}
\dot{x}_1 &= v \cos \theta \\
\dot{y}_1 &= v \sin \theta \\
\dot{\theta} &= \omega \\
\dot{\omega} &= \dot{X}_1 \\
\dot{X}_1 &= \dot{X}_2
\end{align*}
\]  

(12)

With the chosen flat outputs, the state variables can be expressed as

\[
(x, y_i) = (y_1, y_2)
\]

\[
\theta = \arctan \left( \frac{\dot{y}_2}{\dot{y}_1} \right)
\]

\[
v = \sqrt{\dot{y}_1^2 + \dot{y}_2^2}
\]

\[
\omega = \frac{\dot{\dot{y}}_1 \dot{y}_2 - \dot{\dot{y}}_2 \dot{y}_1}{\dot{y}_1^2 + \dot{y}_2^2}
\]

\[
X_1 = \frac{\dot{y}_1 \dot{y}_1 + \dot{y}_2 \dot{y}_2}{\sqrt{\dot{y}_1^2 + \dot{y}_2^2}}
\]

(13)

The two inputs are then calculated as

\[
\begin{align*}
X_2 &= (-\sin \theta \cdot \dot{y}_1 + \cos \theta \cdot \dot{y}_2 - 2X_1\omega) / v \\
X_3 &= \cos \theta \cdot \dot{y}_1 + \sin \theta \cdot \dot{y}_2 + v \omega^2
\end{align*}
\]  

(14)

With Eqs. (11), (13), and (14), the control inputs \(K\) can be expressed with the flat outputs and their derivatives, given by

\[
K = K(\dot{y}_1, \dot{y}_2, \dot{y}_1, \dot{y}_2, \dot{y}_1, \dot{y}_2) = A^{-1} \begin{bmatrix} X_1 \\ X_2 \end{bmatrix} - BF
\]  

(15)

Note that \(K\) should be non-negative. Furthermore, when the angular speed \(\dot{\theta}\) or \(\dot{\theta}\) is close to zero, \(A^{-1}\) does not exist and \(K\) cannot be obtained. Under this condition, the control law with Eq. (7) is not used and \(K\) is set to zero, which increases the speed. The above equation can be used in planning a desired trajectory with the passive inputs.

### IV. Trajectory Planning

In trajectory planning of the passive robot [12,13], one requires to find a smooth trajectory \((y_1(t), y_2(t))\) that passes from the initial points to the target point during the time \(t_f\), while the robot only utilizes the passive control torques for moving on the trajectory. To perform the trajectory planning, the boundary conditions of \((y_1(t), y_2(t))\) are first calculated from the initial and final points. Then, a set of mode functions are chosen for fitting the trajectory that can pass through these two end points. Finally, we find a trajectory by solving the nonlinear constrained equations.

The initial conditions \(y_1(0), \dot{y}_1(0), y_2(0), \dot{y}_2(0)\) and the final conditions \(y_1(t_f), \dot{y}_1(t_f), y_2(t_f), \dot{y}_2(t_f)\) of the trajectory can be obtained from the initial states \(x_0, y_e(0), \theta(0), \nu(0), \omega(0)\) and the final states \(x_1(t_f), y_e(t_f), \theta(t_f), \nu(t_f), \omega(t_f)\) with the following expressions

\[
y_1 = x_1, \dot{y}_1 = v \cos \theta, \dot{x}_1 = \chi_1 \cos \theta - v \sin \theta \omega
\]

\[
y_2 = x_2, \dot{y}_2 = v \sin \theta, \dot{x}_2 = \chi_1 \sin \theta + v \cos \theta \omega
\]  

(16)

Note that the above expressions are used for calculating the values of \(\dot{y}_1(0), \dot{y}_2(0), \dot{y}_1(t_f), \dot{y}_2(t_f)\) but the values of the parameters \(X_1(0), X_1(t_f)\) are still needed. Since \(X_1(0), X_1(t_f)\) are not specified in the planning, these values can be arbitrarily selected.

The trajectory is then fitted with the following form

\[
y_1(t) = \Phi_1(t) + a_{\phi_1}(t) + \sum_{i=1}^{k} a_i \phi_i(t) + a_{\phi_2}(t_f - t)
\]

\[
y_2(t) = \Phi_2(t) + b_{\phi_1}(t) + \sum_{i=1}^{k} b_i \phi_i(t) + b_{\phi_2}(t_f - t)
\]  

(17)

where \(\Phi_1, \Phi_2, \phi_i\) are the mode functions and \(k\) the mode number. \(\Phi_1(t), \Phi_2(t)\) is a trajectory that passes through the initial and final points. Here, \(\phi_i(t)\) are the functions with \(\phi_0(0), \phi_0(t_f), \phi_1(t_f), \phi_2(t_f)\) zero. In this paper, \(\phi_j(t), j=1,2\) are chosen to be the following polynomial functions of time [15]

\[
\Phi_j(t) = c_{j0} + c_{j1} t + c_{j2} t^2 + c_{j3} t^3 + c_{j4} t^4 (t - t_f)
\]

\[
+ c_{j5} t^5 (t - t_f)^2, j=1,2
\]  

(18)

The coefficients \(c_{jk}\) \((j=1,2, k=1,\ldots, 5)\) are solved using the given boundary conditions \(y_1(0), \dot{y}_1(0), \dot{y}_2(0), y_1(t_f), \dot{y}_1(t_f), \dot{y}_2(t_f)\). With the choice of \(\Phi_j(t), c_{jk}\) are easily obtained as
Furthermore, the mode functions \( \phi_i(t) \) are selected as [15]

\[
\phi_i(t) = t^{i+3}(t - t_f)^3
\]  

These mode functions possess the property of having zero derivatives up to the order 2 at 0 and \( t_f \).

Once the flat outputs are parameterized with Eq. (17), the states and inputs are also represented as functions of \( a, b, \chi_1(0), \chi_1(t_f) \). Trajectory generation then can be achieved by solving the following nonlinear constrained optimization problem.

\[
\begin{align*}
\min_{a,b,\chi_1(0),\chi_1(t_f)} & \quad J = \int_0^T \left( [q^TV^T]Q[q^TV^T]^T + \tau^TR\tau \right) dt \\
\text{subject to} & \quad K(a, b, \chi_1(0), \chi_1(t_f)) \geq 0, \\
& \quad -\tau_m \leq \tau(a, b, \chi_1(0), \chi_1(t_f)) \leq \tau_m
\end{align*}
\]  

where \( J \) is user-defined objection function, \( Q, R \) are positive definite symmetric weight matrices, and \( \tau_m \) the maximum torque of the brake motor. Once the desired trajectory is found, the brake torque can be obtained in the control using the structure of the passive robot.

V. MODEL PREDICTIVE CONTROL

To implement this structure within the application of passive robot, model predictive control approach is utilized [16-18], as shown in Fig. 3. In trajectory planning based on MPC, given the target point, the currently measured states and user-applied forces, a feasible trajectory and the passive inputs are predicted for the period \( T \). Then, predictive passive inputs are used for controlling the robot while it is pushed by the user. Since MPC is based on a constant value of currently measured user-applied force, while in reality user-applied force are time varying, the MPC only uses the predictive inputs to control the robot over a period \( \Delta t \) (\( \Delta t < T \)). At time \( \Delta t \), MPC needs to read the states and user-applied forces, and make a new trajectory for the period \( T - \Delta t \) again. The procedure will continue until the period is non-positive.

Note that, the optimization problem with Eq. (21) should be solved at every time step. To reduce the computation time, we do not find the optimal solution, but search for a feasible
solution that can satisfy the constraints.

VI. SIMULATION RESULTS

In order to demonstrate the effectiveness of the proposed approach, the MPC was applied to a guidance problem of the passive walker. The parameter values of the robot in the simulation are $m=18.2$ kg, $I=0.4$ kgm$^2$, $r=0.0825$ m, and $b=0.1624$ m. The start and end points $(x_c, y_c, \theta)$ are $(0 \text{ m}, 0 \text{ m}, \pi/2 \text{ rad})$ and $(6 \text{ m}, 6 \text{ m}, \pi/2 \text{ rad})$, respectively. The condition of steady state walking is first considered, the speed and angular speed $(v, \omega)$ of the starting and end points are thus set to the same values $(0.15 \text{ m/s}, 0 \text{ rad/s})$. The time parameter $t_f$ is set as 60 s. The user-applied force is assumed as a function of time $20+3\sin(\pi/2)t$. Fig. 4 shows the actual and planned trajectories with $\Delta t=0.2$ s. We found that the computed trajectory is almost along the initial path of the planned trajectory in each time step and approaches the target progressively. The trajectories of $x, y, \theta, v, \omega$ and the values of the parameters $K_r, K_i$ are shown in Figs. 5 and 6, respectively. Because the user-applied force are assumed to have oscillations, the speed $v$ and the parameters $K_r, K_i$ also show oscillations.

Since the time parameter $\Delta t$ affects the efficiency of MPC, simulations with different values of $\Delta t$ are performed. Fig. 7 shows the trajectories generated with the values of $\Delta t$ as 0.2, 0.5, 1 and 2 s. These trajectories are slightly different, but can all reach the target. Table I shows the errors between the desired and actual states at end point. The results show that the smaller errors are obtained with smaller $\Delta t$ values. However, a small $\Delta t$ value requires a large number of time steps resulting in low efficiency. To deal with this problem,

<table>
<thead>
<tr>
<th>$\Delta t$ (sec)</th>
<th>$x$ error (m)</th>
<th>$y$ error (m)</th>
<th>$\theta$ error (rad)</th>
<th>$v$ error (m/s)</th>
<th>$\omega$ error (rad/s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.2</td>
<td>2.93e-7</td>
<td>2.44e-4</td>
<td>1.75e-5</td>
<td>3.37e-3</td>
<td>3.57e-4</td>
</tr>
<tr>
<td>0.5</td>
<td>2.30e-7</td>
<td>2.33e-3</td>
<td>2.26e-6</td>
<td>1.22e-2</td>
<td>3.04e-4</td>
</tr>
<tr>
<td>1</td>
<td>4.36e-7</td>
<td>6.64e-5</td>
<td>3.38e-5</td>
<td>2.99e-2</td>
<td>3.27e-5</td>
</tr>
<tr>
<td>2</td>
<td>3.54e-5</td>
<td>2.52e-2</td>
<td>1.54e-3</td>
<td>4.56e-3</td>
<td>8.26e-5</td>
</tr>
<tr>
<td>$2&amp;0.2$</td>
<td>1.47e-8</td>
<td>2.46e-4</td>
<td>7.30e-7</td>
<td>3.38e-3</td>
<td>1.71e-5</td>
</tr>
</tbody>
</table>

Fig. 7. The trajectories of the different $\Delta t$ values in MPC.

Fig. 8. The actual trajectories in MPC of the passive robot.

Fig. 9. The state variables $x, y, \theta, v, \omega$ in MPC of the passive robot.

Fig. 10. The values of $K_r$ and $K_i$ in MPC of the passive robot.
we propose using a large $\Delta t$ value early on in the path followed by smaller $\Delta t$ value when the robot is nearer to the end point. In Table I, the $\Delta t$ value 2&0.2 represents that the first $\Delta t$ value is 2 s over $t=0[58]$ and the second $\Delta t$ value is 0.2 s over $t=[58, 60]$. The simulation results show that accurate results are obtained with $\Delta t$ values 2&0.2.

The static conditions at the initial and final points are further considered in the guidance of the passive robot. In this simulation, the speed and angular speed ($v$, $\omega$) of the starting and end points are set at small values ($10\text{-}3 \text{ m/s}, 0 \text{ rad/s}$). In the trajectory planning, we plan the path with three sub-paths: a straight line for accelerating the heading speed to walking speed 0.15 m/s, a path of steady state walking, and a straight line for decelerating the heading speed to a low speed 1.0e-3 m/s. Because it is difficult to decelerate the heading speed to a low speed when the user also applies the force, the user-applied force is assumed to be zero in the deceleration path. In this simulation, two intermediate points are added and their state $(x, y, \theta, v, \omega)$ values are $(0, 0.15, \pi/2, 0.15, 0)$ and $(6, 5.85, \pi/2, 0.15, 0)$. The time to reach these two points is set as 2 and 58 s, respectively. Figs. 8 and 9 show the path and the trajectories with $\Delta t=0.2$ s, respectively. The values of the parameters $K_a$, $K_i$ are shown in Fig. 10. The results show that the trajectory of the speed $v$ consists of acceleration, constant speed with oscillation, and deceleration. Furthermore, since the initial speed is low, high values of parameters $K_a$, $K_i$ are obtained in the acceleration path.

VII. CONCLUSION

In this paper, we have presented a novel method for trajectory planning of the time-varying gains of a braking control law for a passive robot walking helper. The dynamic model of the vehicle with braking control law was first established. The trajectories of the gains were developed by using approaches from the theory of differential flatness and dynamic feedback linearization. Accordingly, feasible trajectories were found by solving a nonlinear program. Furthermore, model predictive control was also implemented for the passive robot walking helper since user applied forces are not known prior. The simulation results show that the passive robot walking helper with model predictive control can accurately guide the user to a goal, demonstrating the effectiveness of the proposed control scheme.

REFERENCES


