The 2010 IEEE/RSJ International Conference on
Intelligent Robots and Systems
October 18-22, 2010, Taipei, Taiwan

Control and Path Planning of a Walk-Assist Robot using Differential Flatness

Chun-Hsu Ko and Sunil K. Agrawal

Abstract—With the growth of elderly population in our society, technology will play an important role in providing functional mobility to humans. In this paper, we propose a robot walking helper with both passive and active control modes of guidance. From the perspective of human safety, the passive mode adopts the braking control law on the wheels to differentially steer the vehicle. The active mode can guide the user efficiently when the passive control with user-applied force is not adequate for guidance. The theory of differential flatness is used to plan the trajectory of control gains within the proposed scheme of the controller. Since the user input force is not known a-priori, the theory of model predictive control is used to periodically compute the trajectory of these control 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 effectively.

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. To enhance their safety and convenience, intelligent walking aids are needed. Robotics contains several useful technology components, e.g., sensing, actuation, motion control, and computer 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 [1,5,7-10], obstacle avoidance [2,8], and health monitoring [1,7] have been developed. To control a robot walking helper, Spenko et al. [7] used a variable damping model to increase the walking stability. Hirata et al. [8] proposed an adaptive motion control algorithm for obstacle/step avoidance and gravity compensation. Chuy et al. [2] used the passive behavior to enhance the interaction between the user and the active support system. Agrawal et al. [10] proposed a two-phase passive control algorithm for guiding a user to attain desired position and orientation, while allowing for small errors.

In general, the robot walking helpers can be classified into these 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 supplied forces. Controlled brakes are used to steer the walker while constantly extracting energy from the system. With this property of dissipation of energy, they are inherently safer and tend to avoid build up of energy.

However, if a user would like to increase the walking speed, steer the helper, or walk uphill with the walker, large user-applied forces are needed. Furthermore, passive mode may not be enough to achieve effective trajectory planning and control. Hence, in order to effectively help a user, it is important to consider both active and passive control strategies with user applied forces to steer to the goal.

In this paper, we propose using passive and active control modes in effective guidance of robot walking helper. Passive mode, with braking control law on the wheels to differentially steer the vehicle, is first utilized. When the passive control with user-applied forces is not sufficient for accurate guidance, active mode is then used. Both with passive and active modes, the robot walker is controlled to maintain a constant walking speed. The method of differential flatness is used to select time varying trajectories of the control 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 control law is explored to seek the property of differential flatness. The states and control 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. With differential flatness approach, a feasible trajectory with passive or active control mode can be easily found by using the nonlinear programming. Since the user applied forces are not known apriori, model predictive control (MPC) is used to find the solution, repetitively.

The remainder of this paper is organized as follows: Section II describes the dynamic equations of the robot walking helper. In Section III, differential flatness approach is addressed. Trajectory planning and model predictive control are described in Sections IV and V. Section VI presents the simulation results Finally, concluding remarks are given in Section VII.

This work was supported in part by the National Science Council of Taiwan, ROC. under the Grant NSC 98-2918-I-214-001 and NSC 96-2628-E-009-164-MY3. The second author also gratefully acknowledges partial support from World Class University program.

C. H. Ko is with the Department of Electrical Engineering, I-Shou University, Kaohsiung, Taiwan ROC (e-mail: chko@isu.edu.tw).

Sunil K. Agrawal is with the Department of Mechanical Engineering, University of Delaware, Newark, USA (corresponding author; phone: 302-831-8049; fax: 302-831-3619; e-mail: agrawal@udel.edu).

978-1-4244-6676-4/10/$25.00 ©2010 IEEE

Fig. 1. A robot walking helper.

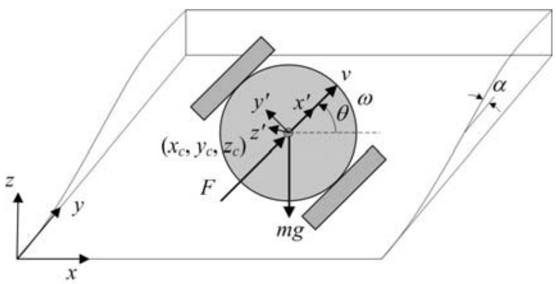

Fig. 2. The configuration of a robot walking helper.

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 driven by motors with encoders, two passive casters, an ultrasonic sensor array, a force sensor, and a controller. The motors provide the proper torques for controlling the motion of the robot walking helper. The encoders are used to measure the wheel speed, while the force sensor detects the user applied force. To design the controller, a simplified dynamic model is described next.

Fig. 2 shows the robot on a slope in world coordinates xyz, [10,12] given by

$$q=[x_c, y_c, z_c, \theta]^T, \tag{1}$$

where x_c, y_c, z_c are the coordinates of the center of mass and θ is the heading angle of the robot. The slope angle is assumed to be a function of y_c, $\alpha=\alpha(y_c)$. When the robot moves on the slope, the body coordinates $x'y'z'$ rotate θ about z axis and α about the x axis. The rotation matrix R is calculated as

$$R=\begin{bmatrix} \cos\theta & -\sin\theta & 0 \\ \cos\alpha\sin\theta & \cos\alpha\cos\theta & -\sin\alpha \\ \sin\alpha\sin\theta & \sin\alpha\cos\theta & \cos\alpha \end{bmatrix} \tag{2}$$

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

$$\begin{bmatrix} \dot{x}_c \\ \dot{y}_c \\ \dot{z}_c \end{bmatrix} = v\hat{x}' = v\begin{bmatrix} \cos\theta \\ \sin\theta\cos\alpha \\ \sin\theta\sin\alpha \end{bmatrix} \tag{3}$$

and

$$\dot{\theta}=\omega \tag{4}$$

where v is the heading speed and ω the turning speed. The angular velocity can be expressed as

$$\vec{\omega}=\dot{\alpha}\hat{x}+\omega\hat{z}'=[\dot{\alpha} \quad -\omega\sin\alpha \quad \omega\cos\alpha]^T \tag{5}$$

where $\dot{\alpha}=\alpha'(y_c)\dot{y}_c$. The equations of motion with the user-applied force F, motor applied torques, the gravity force on a slope and the no-slip constraint force/moment are written as

$$m[\ddot{x}_c \quad \ddot{y}_c \quad \ddot{z}_c]^T=(F+\frac{\tau_r+\tau_l}{r})\hat{x}'-mg\hat{z}+\lambda\hat{y}'+p\hat{z}' \tag{6}$$

and

$$I\dot{\vec{\omega}}+\vec{\omega}\times I\vec{\omega}=\frac{b(\tau_r-\tau_l)}{r}\hat{z}'+M_x\hat{x}'+M_y\hat{y}' \tag{7}$$

Here, m is the robot mass, I the moment of inertia of the robot, r is the wheel radius, b half distance between the two wheels, τ_r and τ_l the motor torques on the right and left wheels, λ the constraint force, p the normal reaction force and M_x, M_y the constraint moments. The moment of inertia I can be calculated as

$$I=RI_{body}R^T, I_{body}=\begin{bmatrix} I_{11} & I_{12} & I_{13} \\ I_{12} & I_{22} & I_{23} \\ I_{13} & I_{23} & I_{33} \end{bmatrix} \tag{8}$$

With the assumption of symmetry about the plane $x'z'$, the values of I_{12} and I_{23} are zeros. By differentiating Eq. (3), $[\ddot{x}_c, \ddot{y}_c, \ddot{z}_c]^T=v\dot{\hat{x}}'+\dot{v}\hat{x}'$. On substituting this $[\ddot{x}_c, \ddot{y}_c, \ddot{z}_c]^T$ into Eq. (6) and pre-multiplying by $\hat{x}'^T$, we get

$$m\dot{v}=F+\frac{\tau_r+\tau_l}{r}-mg\sin\theta\sin\alpha \tag{9}$$

Furthermore, by differentiating Eq. (5),

$$\dot{\vec{\omega}}=[\ddot{\alpha} \quad -\dot{\omega}\sin\alpha-\omega\cos\alpha\dot{\alpha} \quad \dot{\omega}\cos\alpha-\omega\sin\alpha\dot{\alpha}] \tag{10}$$

On substituting Eq. (10) into Eq. (7) and pre-multiplying by $\hat{z}'^T$, we get

$$I_{33}\dot{\omega}=\frac{b(\tau_r-\tau_l)}{r}-\dot{\alpha}^2\cos\theta\sin\theta(I_{11}-I_{22})-\dot{\alpha}\sin\theta\omega I_{13} \tag{11}$$

From Eqs. (3), (4), (9), and (11), the state equations of the robot on a slope are expressed as

$$\begin{aligned} \dot{x}_c &= v\cos\theta \\ \dot{y}_c &= v\sin\theta\cos\alpha \\ \dot{z}_c &= v\sin\theta\sin\alpha \\ \dot{\theta} &= \omega \\ \dot{v} &= \frac{\tau_r+\tau_l}{mr}+\frac{F}{m}-g\sin\theta\sin\alpha \\ \dot{\omega} &= \frac{b(\tau_r-\tau_l)}{I_{33}r}-\dot{\alpha}^2\cos\theta\sin\theta\frac{I_{11}-I_{22}}{I_{33}}-\dot{\alpha}\sin\theta\omega\frac{I_{13}}{I_{33}} \end{aligned} \tag{12}$$

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

$$\tau_r = -K_r \dot{\theta}_r, \quad \tau_l = -K_l \dot{\theta}_l \tag{13}$$

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

$$\dot{\theta}_r = (v + b\omega)/r, \quad \dot{\theta}_l = (v - b\omega)/r \tag{14}$$

Substituting Eqs. (13) and (14) into Eq. (12), we can obtain the dynamic equations of the robot given by

$$\begin{aligned} \dot{q} &= SV \\ \dot{V} &= AK + B \end{aligned} \tag{15}$$

where

$$\begin{aligned} V &= \begin{bmatrix} v \\ \omega \end{bmatrix}, S = \begin{bmatrix} \cos\theta & 0 \\ \sin\theta\cos\alpha & 0 \\ \sin\theta\sin\alpha & 0 \\ 0 & 1 \end{bmatrix}, \\ A &= \begin{bmatrix} -\dfrac{v+b\omega}{mr^2} & -\dfrac{v-b\omega}{mr^2} \\ -\dfrac{b(v+b\omega)}{I_{33}r^2} & \dfrac{b(v-b\omega)}{I_{33}r^2} \end{bmatrix}, K = \begin{bmatrix} K_r \\ K_l \end{bmatrix}, \\ B &= \begin{bmatrix} F/m - g\sin\theta\sin\alpha \\ -\dot{\alpha}^2\cos\theta\sin\theta\dfrac{I_{11}-I_{22}}{I_{33}} - \dot{\alpha}\sin\theta\omega\dfrac{I_{13}}{I_{33}} \end{bmatrix} \end{aligned} \tag{16}$$

The entries of 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. If non-negative values of the K are chosen, the robot walker is controlled with a dissipative/passive mode. Otherwise, an active mode is used in controlling the robot walker. 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. Differential Flatness

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 robot system is differentially flat with the flat outputs $(y_1, y_2)=(x_c, y_c)$. With a given slope height $z(y_2)$, the slope angle and its derivatives can be expressed as

$$\begin{aligned} \alpha &= \tan^{-1} z'(y_2) \\ \dot{\alpha} &= \frac{z''}{1+z'^2}\dot{y}_2 \\ \ddot{\alpha} &= \frac{z'''\dot{y}_2^2 + z''\ddot{y}_2}{1+z'^2} - \frac{2z'z''^2\dot{y}_2^2}{(1+z'^2)^2} \end{aligned} \tag{17}$$

Define s as the length of the slope, then

$$\begin{aligned} \dot{s} &= \dot{y}_2/\cos\alpha \\ \ddot{s} &= \frac{\ddot{y}_2 + \dot{y}_2 z'\dot{\alpha}}{\cos\alpha} \\ \dddot{s} &= \frac{\dddot{y}_2 + 2\ddot{y}_2 z'\dot{\alpha} + 2\dot{y}_2 z'^2\dot{\alpha}^2 + \dot{y}_2\dot{\alpha}^2 + \dot{y}_2 z'\ddot{\alpha}}{\cos\alpha} \end{aligned} \tag{18}$$

and

$$\begin{aligned} \theta &= \arctan(\frac{\dot{s}}{\dot{y}_1}) \\ v &= \sqrt{\dot{y}_1^2 + \dot{s}^2} \\ \omega &= \dot{\theta} = \frac{\dot{y}_1\ddot{s} - \ddot{y}_1\dot{s}}{\dot{y}_1^2 + \dot{s}^2} \\ \dot{v} &= \frac{\dot{y}_1\ddot{y}_1 + \dot{s}\ddot{s}}{\sqrt{\dot{y}_1^2 + \dot{s}^2}} \\ \dot{\omega} &= (-\sin\theta\cdot\dddot{y}_1 + \cos\theta\cdot\dddot{s} - 2\dot{v}\omega)/v \end{aligned} \tag{19}$$

With Eqs. (16-19), the control inputs K can be expressed with the flat outputs and their derivatives, given by

$$K = K(\dot{y}_1, \ddot{y}_1, \dddot{y}_1, y_2, \dot{y}_2, \ddot{y}_2, \dddot{y}_2) = A^{-1}\left(\begin{bmatrix} \dot{v} \\ \dot{\omega} \end{bmatrix} - B\right) \tag{20}$$

The above equation can be used in planning a desired trajectory with the active or passive modes.

IV. Trajectory Planning

To perform the trajectory planning [12,13], the boundary conditions of $(y_1(t), y_2(t))$ are firstly calculated from the initial and final points. Then, a set of 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 $y_1(0), \dot{y}_1(0), \ddot{y}_1(0), y_2(0), \dot{y}_2(0), \ddot{y}_2(0)$ and the final conditions $y_1(t_f), \dot{y}_1(t_f), \ddot{y}_1(t_f), y_2(t_f), \dot{y}_2(t_f), \ddot{y}_2(t_f)$ of the trajectory can be obtained with the given slope height $z(y_2)$, the initial states $x_c(0), y_c(0), \theta(0), v(0), \omega(0)$, and the final states $x_c(t_f), y_c(t_f), \theta(t_f), v(t_f), \omega(t_f)$ by using the following expressions

$$\begin{aligned} & y_1 = x_c, \dot{y}_1 = v\cos\theta, \ddot{y}_1 = \dot{v}\cos\theta - v\sin\theta\omega \\ & y_2 = y_c, \dot{y}_2 = v\sin\theta\cos\alpha, \\ & \ddot{y}_2 = \dot{v}\sin\theta\cos\alpha + v\cos\theta\omega\cos\alpha - v\sin\theta\sin\alpha\alpha'\dot{y}_2 \end{aligned} \tag{21}$$

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

The trajectory is then fitted with the following form

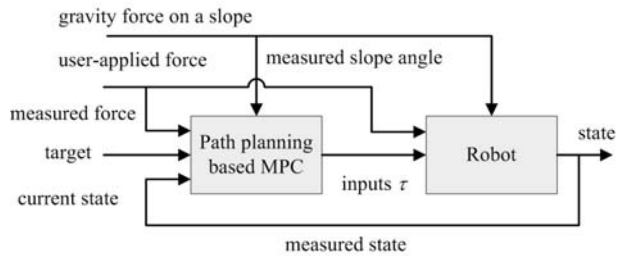

Fig. 3. Model predictive control of the robot.

$$
\begin{aligned}
y_1(t) &= \Phi_1(t) + \sum_{i=1}^{k} a_i \phi_i(t) \\
y_2(t) &= \Phi_2(t) + \sum_{i=1}^{k} b_i \phi_i(t)
\end{aligned}
\tag{22}
$$

where Φ_1, Φ_2, ϕ_i are the mode functions and k the mode number. Here, $\Phi_1(t), \Phi_2(t)$ is a trajectory that passes through the initial and final points. Moreover, $\phi_i(t)$ are the functions with $\phi_i(0), \dot{\phi}_i(0), \ddot{\phi}_i(0), \phi_i(t_f), \dot{\phi}_i(t_f), \ddot{\phi}_i(t_f)$ zero. In this paper, $\Phi_j(t), j=1,2$ are chosen to be the following polynomial functions of time [15]

$$
\begin{aligned}
\Phi_j(t) &= c_{j0} + c_{j1}t + c_{j2}t^2 + c_{j3}t^3 + c_{j4}t^3(t-t_f) \\
&\quad + c_{j5}t^3(t-t_f)^2,\ j=1,2
\end{aligned}
\tag{23}
$$

The coefficients c_{jk} (j=1,2 ,k=1,…,5) are solved using the given boundary conditions $y_j(0), \dot{y}_j(0), \ddot{y}_j(0), y_j(t_f), \dot{y}_j(t_f), \ddot{y}_j(t_f)$. With the choice of $\Phi_j(t)$, c_{jk} are easily obtained as

$$
\begin{aligned}
& c_{j0} = y_j(0), c_{j1} = \dot{y}_j(0), c_{j2} = 0.5\ddot{y}_j(0) \\
& c_{j3} = (y_j(t_f) - c_{j0} - t_f c_{j1} - t_f^2 c_{j2}) / t_f^3 \\
& c_{j4} = (\dot{y}_j(t_f) - c_{j1} - 2t_f c_{j2} - 3t_f^2 c_{j3}) / t_f^3 \\
& c_{j5} = (0.5\ddot{y}_j(t_f) - c_{j2} - 3t_f c_{j3} - 3t_f^2 c_{j4}) / t_f^3
\end{aligned}
\tag{24}
$$

Furthermore, the mode functions $\phi_i(t)$ are selected as [15]

$$
\phi_i(t) = t^{i+3}(t-t_f)^3 \tag{25}
$$

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. (22), the states and inputs are also represented as functions of $a, b, \dot{v}(0), \dot{v}(t_f), t_f$. Trajectory generation then can be achieved by solving a nonlinear constrained optimization problem. With the active mode, the optimization problem is described as

$$
\begin{aligned}
& \min_{a,b,\dot{v}(0),\dot{v}(t_f),t_f} J \\
& \text{subject to} - \tau_m \le \tau(a,b,\dot{v}(0),\dot{v}(t_f),t_f) \le \tau_m
\end{aligned}
\tag{26}
$$

where J is user-defined objection function and τ_m the maximum torque of the brake motor. Furthermore, the optimization problem with the passive mode includes the constraints of the non-negative values in K, expressed as

$$
\begin{aligned}
& \min_{a,b,\dot{v}(0),\dot{v}(t_f),t_f} J \\
& \text{subject to } K(a,b,\dot{v}(0),\dot{v}(t_f),t_f) \ge 0 \\
& \qquad -\tau_m \le \tau(a,b,\dot{v}(0),\dot{v}(t_f),t_f) \le \tau_m
\end{aligned}
\tag{27}
$$

Once the desired trajectory is found, the torques can be obtained for controlling the robot to achieve the target.

V. Model Predictive Control

To implement this structure within the application of robot walking helper, 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, user-applied forces and slope angles, a feasible trajectory and the control inputs are predicted for a period T. 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 Δt ($\Delta t < T$). After time Δt, MPC needs to read the states and user-applied forces, and make a new trajectory for the next period again. The procedure will continue until the stop criterions are satisfied.

Note that a fitted slope function $z(y)$ is used for MPC in each period. Given the current position (y_{c1}, z_{c1}), the target position (y_{c2}, z_{c2}), the measured current slope angle α_1, and the target slope angle α_2, a predictive slope function $z(y)$ is then fitted with the following form

$$
\begin{aligned}
& z(y_c) = d_0 + d_1(y_c - y_{c1}) + d_2(y_c - y_{c1})^2 \\
& + d_3(y_c - y_{c1})^3
\end{aligned}
\tag{28}
$$

With the boundary conditions $z(y_{c1})=z_{c1}$, $z'(y_{c1})=\tan(\alpha_1)$, $z(y_{c2})=z_{c2}$, and $z'(y_{c2})=\tan(\alpha_2)$, the coefficients d_i (i=0,…,3) are solved as

$$
\begin{aligned}
& d_0 = z_1, d_1 = \tan\alpha_1 \\
& d_2 = 3(z_{c2} - z_{c1})(y_{c2} - y_{c1})^{-2} \\
& \qquad - (2\tan\alpha_1 + \tan\alpha_2)(y_{c2} - y_{c1})^{-1} \\
& d_3 = -2(z_{c2} - z_{c1})(y_{c2} - y_{c1})^{-3} \\
& \qquad + (\tan\alpha_1 + \tan\alpha_2)(y_{c2} - y_{c1})^{-2}
\end{aligned}
\tag{29}
$$

To assist the user in walking stably, the passive mode is first selected to plan the trajectory from the present state to the goal point by solving Eq. (27). If no feasible solution can be found, the heading angle at the final point is set as a design variable in Eq. (27). However, if no feasible solution can be found again, the active mode is used to plan the trajectory with Eq. (26). With above procedures, the passive mode can be used as much as possible in the guidance.

VI. Simulation Results

In order to demonstrate the effectiveness of the proposed approach, the MPC was applied to a guidance problem of the walker. The parameter values of the robot in simulations are m=25 kg, I_{11}=2 kgm^2, I_{22}=1.5 kgm^2, I_{33}=3 kgm^2, I_{13}=0.2 kgm^2, r=0.075 m, and b=0.4 m. The start and end points (x_c, y_c, θ)

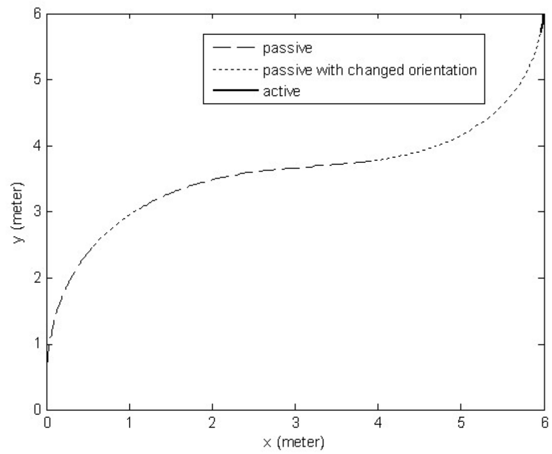

Fig. 4. The trajectory of the robot with f_m=10 N.

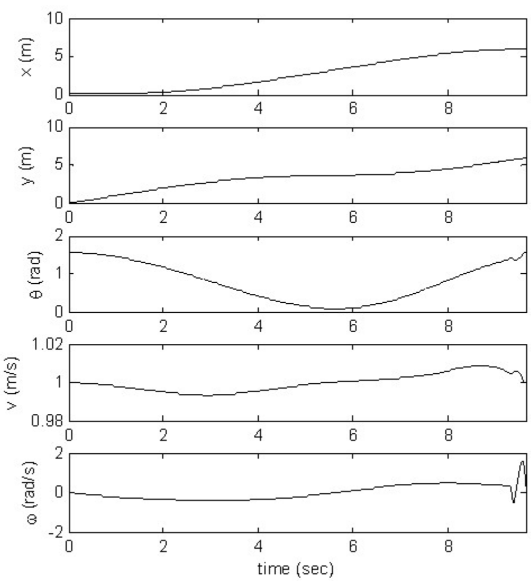

Fig. 5. The state variables x, y, θ, v, and ω with f_m=10 N.

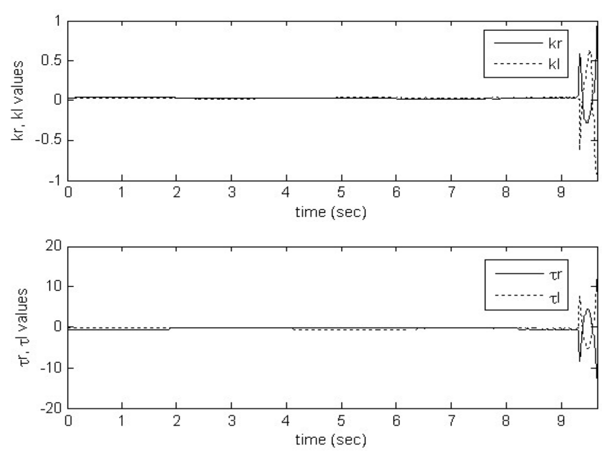

Fig. 6. The values of K_r, K_l, τ_r, and τ_l with f_m=10 N.

are (0 m, 0 m, π/2 rad) and (6 m, 6 m, π/2 rad), respectively. The condition of stable walking is first considered, i.e., the speed and angular speed (v, ω) of the start and end points are

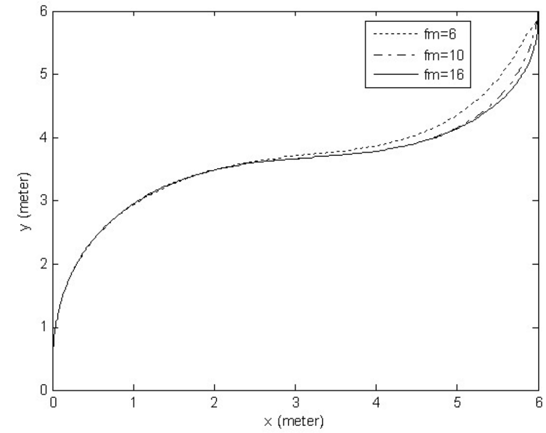

Fig. 7. The trajectories of the robot with f_m=6, 10, and 16 N.

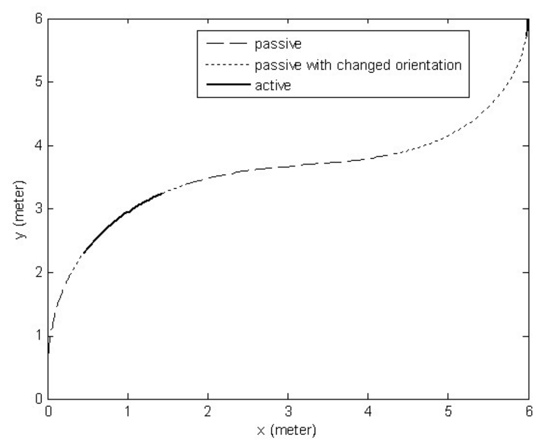

Fig. 8. The trajectory of the robot on a slope with f_m=16 N and h=0.2 m.

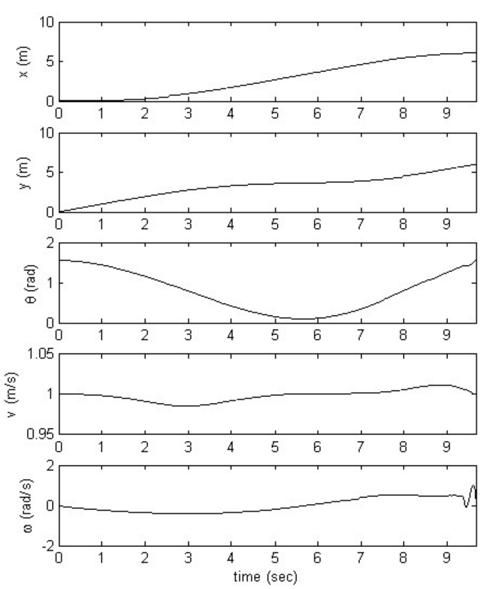

Fig. 9. The state variables x, y, θ, v, and ω of the robot with f_m=16 N and h=0.2 m.

set to the same values (1 m/s, 0 rad/s). The user-applied force is assumed as a function of time $f_m(1+0.2\sin(\pi/2)t)$. Fig. 4 shows the trajectory with f_m=10 N. We observe that the

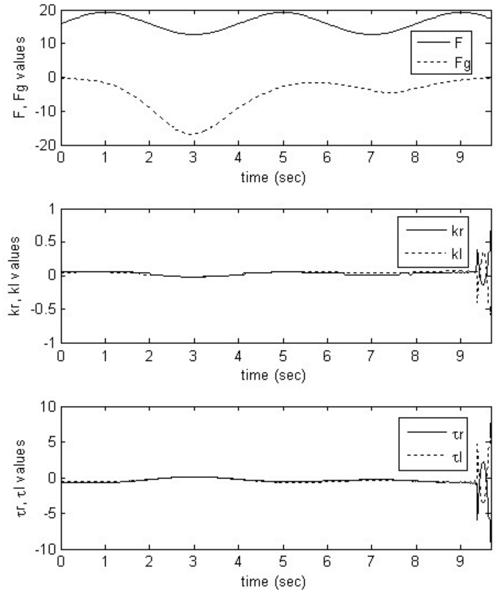

Fig. 10. The values of F, F_g, K_r, K_l, τ_r, and τ_l with f_m=16 N and h=0.2 m.

passive mode is thus mostly used during the maneuver and the active mode is used for approaching the target. The trajectories of x, y, θ, v, ω and the values of K_r, K_l are shown in Figs. 5 and 6, respectively. The robot is almost controlled with a constant speed of 1 m/s. When the robot approaches the goal position, the active mode with a large torque is used for accurately steering the robot to the heading angle of goal point. Fig. 7 shows the trajectories with f_m=6, 10, 16 N, respectively. The results show that the robot reaches the final point accurately with a large force (f_m=16 N). However, with a small force (f_m=6, 10 N), the robot requires the passive/active mode for reaching the goal.

The guidance of the robot on a slope is further considered. The slope height z is assumed to be a function of y, $z(y) = 0.2/(1+e^{-2(y-3)})$. Figs. 8-10 show the trajectories and the values of the parameters K_r, K_l on a slope height h=0.2 m with a user-applied force f_m=16 N. We found that the active mode is used around y=3 m due to a large slope angle and a large gravity force Fg. The positive torques are applied by the robot walker. The proposed passive/active control mode is effective for helping the user walking on the slope with a constant speed.

VII. Conclusion

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

References

[1] H. Yu, M. Spenko, and S. Dubowsky, "An adaptive shared control system for an intelligent mobility aid for the elderly," *Auton. Robots*, vol. 15, no. 1, pp. 53–66, 2003.

[2] O. Chuy, Y. Hirata, Z. Wang, and K. Kosuge, "A control approach based on passive behavior to enhance user interaction," *IEEE Trans. Robotics*, vol. 23, no. 5 pp. 899-908, Oct. 2007.

[3] A. J. Rentschler, R. A. Cooper, B. Blaschm, and M. L. Boninger, "Intelligent walkers for the elderly: Performance and safety testing of VA-PAMAID robotic walker," *J. Rehab. Res. Dev.*, vol. 40, no. 5, pp. 423–432, 2003.

[4] G. Wasson, P. Sheth, M. Alwan, K. Granata, A. Ledoux, and C. Huang, "User intent in a shared control framework for pedestrian mobility aids," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, pp. 2962–2967, 2003.

[5] B. Graf, "An adaptive guidance system for robotic walking aids," *Journal of Computing and Information Technology*, vol. 17, no. 1, pp. 109–120, 2009.

[6] A. M. Sabatini, V. Genovese, and E. Pacchierotti, "A mobility aid for the support to walking and object transportation of people with motor impairments," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, pp. 1349–1354, 2002.

[7] M. Spenko, H. Yu, and S. Dubowsky, "Robotic personal aids for mobility and monitoring for the elderly," *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 14, no. 3, pp. 344-351, Sept. 2006.

[8] Y. Hirata, A. Hara, and. K. Kosuge, "Motion control of passive intelligent walker using servo brakes," *IEEE Trans. Robotics*, vol. 23, no. 5, pp. 981-990 ,Oct. 2007.

[9] N. Nejatbakhsh and K. Kosuge, "Optimal Guidance by Omnidirectional Passive Mobility Aid System," *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 5786-5791, 2006.

[10] J. C. Ryu, K. Pathak, and S. K. Agrawal, "Control of a passive mobility assist robot," *Journal of Medicial Devices*, vol. 2, pp. 011002 (7 pages), March 2008.

[11] S. H. Yu, C. H. Ko, and K. Y. Young, "On the design of a robot walking helper with human intension and envoronmental sensing," in *Proc. CACS Int. Auto. Cont. Conf.*, 2008.

[12] K. Pathak and S. K. Agrawal, "An integrated path-planning and control approach for nonholonomic unicycles using switch local potentials," *IEEE Tran. on Robotics*, vol. 21, no. 6, pp. 1201-1208, December 2005.

[13] J. C. Ryu, S. K. Agrawal, and J. Franch, "Motion planning and control of a tractor with a steerable trailer using differential flatness," *Journal of Computational and Nonlinear Dynamics*, vol. 3, pp. 031003 (8 pages), July 2008.

[14] H. Sira-Ramirez and S. K. Agrawal, *Differential Flat Systems*. 1st ed., New York: Marcel Dekker, 2004.

[15] S. K. Agrawal and T. Veeraklaew, "A higher-order method for dynamic optimization of a class of linear systems," *Journal of Dynamic Systems, Measurement, and Control*, vol. 118, pp. 786-791, December 1996.

[16] M. Benosman and K. Y. Lum, "Online references reshaping and control reallocation for nonlinear fault tolerant control," *IEEE Trans. Cont. Syst. Tech.*, vol. 17, no. 2, pp. 366-379, Mar. 2009.

[17] D. Gu and H. Hu, "Receding horizon tracking control of wheeled mobile robots," *IEEE Trans. Cont. Syst. Tech.*, vol. 14, no. 4, pp. 743-749, July 2006.

[18] L. Magni and R. Scattolini, "Model predictive control of continuous-time nonlinear systems with piecewise constant control," *IEEE Trans. Auto. Cont.*, vol. 49, no. 6, pp. 900-906, June 2004.