Optimal path-following control of a smart powered wheelchair

Nghia Nguyen, Hung T.Nguyen, *Senior Member, IEEE* and Steven Su, *Member, IEEE*

***Abstract*—This paper proposes an optimal path-following control approach for a smart powered wheelchair. Lyapunov's second method is employed to find a stable position tracking control rule. To guarantee robust performance of this wheelchair system even under model uncertainties, advanced robust tracking is utilised based on the combination of a systematic decoupling technique and neural network design. A calibration procedure is adopted for the wheelchair system to improve positioning accuracy. After the calibration, the accuracy is improved significantly. Two real-time experimental results obtained from square tracking and door passing tasks confirm the performance of proposed approach.**

I. INTRODUCTION

POWERED wheelchairs can provide mobility assistance to disabled people. However, for some people with excessively weak residual physical capacities and serious cognitive impairments it may be difficult or impossible to use these wheelchairs with conventional joystick. Therefore, to help these people, smart wheelchairs, which consist of two control levels known as supervisory control level and drive control level, have been the subject of intensive research activities [1-6]. These smart wheelchairs can be designed with intelligent functionalities such as obstacle avoidance, path-following, and docking.

As path-following is one of the main intelligent functionalities of the smart wheelchairs, several control methods have been used to design this feature. In [1], conventional feed-back control is applied for the path-following control of a semi-autonomous wheelchair, while flatness based trajectory tracking control is used in [2] . Both methods in [1, 2] are kinematics based designs. However, since dynamic models of the wheelchairs are not considered, these approaches may not be optimal.

In term of path-following control, the accuracy of the wheelchair position measurement plays an important role. Odometry is one of the most practical and widely used methods for wheelchair positioning due to its ease of use and cost effectiveness. Furthermore, it can provide easily accessible real-time information in between periodic absolute position measurement. The disadvantage of odometry is its unbounded accumulation of errors known as systematic errors, which cause inaccuracy position information.

In this paper, we propose an optimal path-following control method for a smart wheelchair. In the supervisory control level, a stable position tracking control method based on Lyapunov theory in [7] is applied. To improve accuracy in actual wheelchair position, a modified *Bidirectional Square Path* (BSP) strategy is also applied. The robust performance of the drive control is guaranteed even under model uncertainties using our advanced robust tracking control as described in [8]. Since our method takes wheelchair dynamics into account, it can guarantee the path-following feature with desired accuracy and response.

The paper is organized as follows. In the next section, an optimal path-following control structure is proposed. Position control design and systematic error correction are described in the section III. An advanced robust control of the drive level is designed in the section IV. Real-time experimental results are shown in the section V. The conclusion can be found in section VI.

II. PROPOSED OPTIMAL PATH FOLLOWING CONTROL

In the control structure presented in the Figure 1, the path generator block in the supervisory control level generates the reference position. The actual position is calculated from actual velocities, then comparing with its reference to compute the error position. To force this error position converge to zero, a stable position tracking controller is designed via the use of Lyapunov's second method. While an advanced robust control approach is used to guarantee the robust performance in the drive control level.

To simplify the analysis and synthesis of the multivariable wheelchair system, a systematic decoupling technique [9] is used to decompose the wheelchair system into two sub-systems known as linear velocity system and angular velocity system. Two optimal neural network controllers are then designed independently for these sub-systems.

III. POSITION CONTROL DESIGN AND SYSTEMATIC ERROR CORRECTION

A. Position control design

1) Constraint equation

A powered wheelchair system is shown in Figure 2. The linear and angular velocity can be calculated as in (3.1):

This work was supported in part by Australian Research Council under Discovery Grant DP0666942 and LIEF Grant LE0668541.

Nghia Nguyen is with Faculty of Engineering, University of Technology, Sydney, Broadway, NSW 2007, Australia (phone: +612-9514-2451; fax: +61 2 9514 2868; e-mail: tnnguyen@eng.uts.edu.au).

Hung .T. Nguyen is with Faculty of Engineering, University of Technology, Sydney, Broadway, NSW 2007, Australia (e-mail: Hung.Nguyen@uts.edu.au).

Steven Su is with Faculty of Engineering, University of Technology, Sydney, Broadway, NSW 2007, Australia (e-mail: Steven.Su@uts.edu.au).

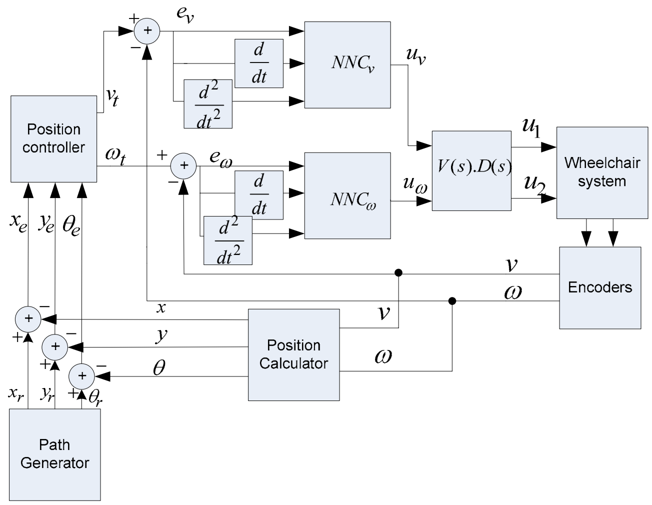

Figure 1: Optimal path following control scheme

$$\begin{cases} v = \dfrac{1}{4}\left(D_r \omega_r + D_l \omega_l\right) \\ \omega = \dfrac{1}{2b}\left(D_r \omega_r - D_l \omega_l\right) \end{cases} \tag{3.1}$$

where D_r and D_l are righ-hand and lelt-hand wheel diameter respectively. Symbol b is defined as the wheelbase as shown in Figure 2. ω_r and ω_l are angular velocity of the right-hand wheel and left-hand wheel.

Under Cartesian coordinate, wheelchair position can be expressed as:

$$q = (x, y, \theta)^T \tag{3.2}$$

where $(x, y)^T$ is the coordinate of the midpoint of rear wheels P_c. θ is the angle of forward direction of the main body with respect with X axis. The superscript T denotes transpose operation.

The constraint equation in the digital form for the wheelchair can be obtained from [6] shown as follows:

$$\begin{cases} x(k+1) = x(k) + Ts \cdot v(k) \cdot \cos(\theta(k)) \\ y(k+1) = y(k) + Ts \cdot v(k) \cdot \sin(\theta(k)) \\ \theta(k+1) = \theta(k) + Ts \cdot \omega(k) \end{cases} \tag{3.3}$$

where *Ts* is the sampling period

2) Position control rule

Define the error position as follows:

$$\begin{pmatrix} x_e \\ y_e \\ \theta_e \end{pmatrix} = \begin{pmatrix} \cos(\theta) & \sin(\theta) & 0 \\ -\sin(\theta) & \cos(\theta) & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x_r - x \\ y_r - y \\ \theta_r - \theta \end{pmatrix} \tag{3.4}$$

The purpose of path-following control is to make the position tracking error converge to zero. This is equivalent to finding a control law to force the tracking error converge to zero. Based on Lyapunov theory, the control rule proposed in [7] is described in the following equation:

$$\begin{pmatrix} v_t \\ \omega_t \end{pmatrix} = \begin{pmatrix} v_r \cos(\theta_e) + K_x x_e \\ \omega_r + v_r \left(K_y y_e + K_\theta \sin(\theta_e)\right) \end{pmatrix} \tag{3.5}$$

where $(v_r \quad \omega_r)^T$ is reference velocity. K_x, K_y and K_θ are positive parameters to be designed [7].

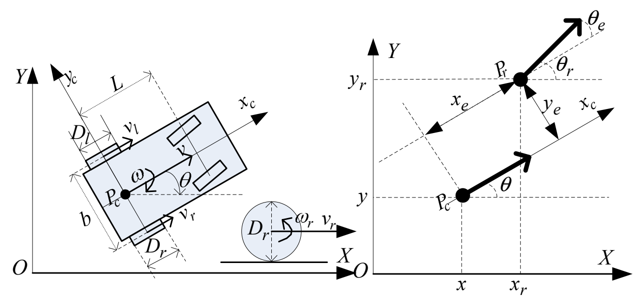

Figure 2: A powered wheelchair system

B. *Systematic error correction*

The wheelchair in this paper uses two optical encoders mounted on the two drive motors to count the wheel revolutions. Using the constraint equation (3.3), it is straight-forward to compute the momentary position of the wheelchair relative to a known starting position. This computation is called odometry [10]. However, there are two error categories known as systematic errors and nonsystematic errors, which both cause inaccuracies in a measured position.

The systematic errors are caused by imperfection in the design and mechanical implementation of the wheelchair such as unequal wheel diameters and effective wheelbase uncertainties. Interaction of the wheelchair with unpredictable feature of environment (slippery or uneven surface) is the main source of nonsystematic errors.

This paper implements the *BSP* in [10] to reduce systematic errors for improving an accuracy of a measured position.

Before the calibration procedure, the two wheel diameters and the wheelbase of the wheelchair are:

$$D_r = 0.35[m]; D_l = 0.35[m]; b = 0.55[m] \tag{3.6}$$

The wheelchair was calibrated with an 8mx8m square path. To avoid slippage, the wheelchair was running in a slow speed, v=0.3 [m/s]. The wheelchair was programmed for 5 runs in clockwise (CW) direction and 5 runs in counterclockwise (CCW) direction.

After this runs, the return error positions were used to calibrate the wheelchair. The updated wheelchair parameters of the wheelchair after calibration procedure were found as following:

$$\begin{cases} D_{ref} = 0.349734[m] \\ D_{lef} = 0.350266[m] \end{cases}; b_{ef} = 0.528661[m] \tag{3.7}$$

Figure 3 shows the trajectory of the wheelchair before and after calibration. It indicates the accuracy of measuring wheelchair position improves significantly after the calibration.

IV. ADVANCED ROBUST CONTROL DESIGN OF THE DRIVE CONTROL LEVEL

Advanced robust tracking control is designed in this section to guarantee the robust performance of the multivariable wheelchair system under model uncertainties.

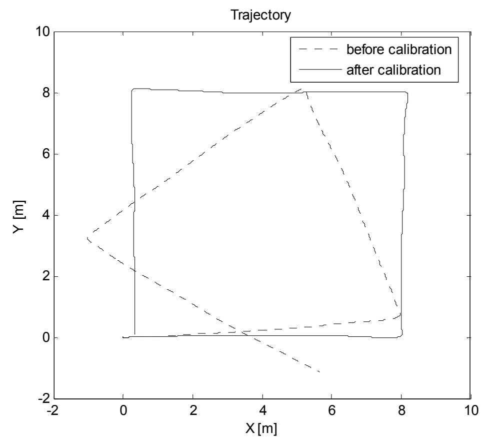

Figure 3: Wheelchair trajectory before and after calibration

The dynamic wheelchair model obtained in [8] is regarded as a multivariable system. This wheelchair model varies from lower-bounded transfer-function matrix $G_1(s)$ to upper-bounded transfer-function matrix $G_2(s)$.

The nominal model of this wheelchair system $G_0(s)$ can be simplified as:

$$G_0(s) = \begin{bmatrix} \dfrac{1.4}{(1+0.8s)(1+0.225s)} & \dfrac{0.125}{(1+0.4s)(1+0.15s)} \\ \dfrac{0.1}{(1+0.35s)(1+0.2s)} & \dfrac{1.8}{(1+0.5s)(1+0.2s)} \end{bmatrix} \quad (4.1)$$

A. Decoupling design of the drive control level

To simplify the control design, decoupling technique is used to cast a multivariable problem into scalar problems. In [8], the triangularization technique is used to construct the desired decoupler D(s) in the following form:

$$D(s) = \begin{bmatrix} 1 & -0.0893\dfrac{(0.8s+1)(0.225s+1)}{(0.4s+1)(0.15s+1)} \\ 0 & 1 \end{bmatrix} \quad (4.2)$$

The obtained decoupled transfer function matrix is triangular-diagonal-dominant (TDD) as follows:

$$P_0(s) = G_0(s)D(s) = \begin{bmatrix} \dfrac{280}{(5+4s)(40+9s)} & 0 \\ \dfrac{10}{(20+7s)(5+s)} & \dfrac{12332(s+2.89)}{(20+7s)(5+s)(2+s)} \end{bmatrix} \quad (4.3)$$

In order to diagonalise the triangular matrix $P(s)$, a pre-compensator V(s) is chosen as follows:

$$V(s) = \begin{bmatrix} 1 & 0 \\ c & 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ -8.11\dfrac{2+s}{100s+289} & 1 \end{bmatrix} \quad (4.4)$$

Then, the diagonalized model of the wheelchair can be obtained in the simplified form as follows:

$$P_D(s) = P_0(s)V(s) \approx \begin{bmatrix} \dfrac{280}{(5+4s)(40+9s)} & 0 \\ 0 & \dfrac{123.32}{(5+s)(2+s)} \end{bmatrix} \quad (4.5)$$

B. Neural network design of the drive control level.

After decoupling, the wheelchair system is decomposed into two scalar systems (4.5), which are needed to design two neural network controllers known as NNC_v and NNC_w, respectively. Both neural network controllers are feed-forward multilayer perceptron (MLP) neural networks.

For a given structure of the NNC_v, the output of the neural network can be calculated as follows:

$$u_v = \sum_{i=1}^{m} f_2(w_i \sum_{j=1}^{n} f_1(\overline{w}_{ij} e_{vj})) \quad (4.6)$$

where $f_1(\cdot)$ and $f_2(\cdot)$ the activation functions of hidden layer and output layer, $\overline{w}$ and w are matrix weights

The objective function is defined as:

$$E = \frac{1}{2}(v_t - v)^2 \quad (4.7)$$

To minimize E, it is necessary to change the weights of a neural network in the direction of negative gradient.

$$\Delta w_i = -\alpha\frac{\partial E}{\partial w_i} = -\alpha\frac{\partial E}{\partial v}\frac{\partial v}{\partial u_v}\frac{\partial u_v}{\partial w_i} \quad (4.8)$$

$$\Delta\overline{w}_{ij} = -\alpha\frac{\partial E}{\partial \overline{w}_{ij}} = -\alpha\frac{\partial E}{\partial v}\frac{\partial v}{\partial u_v}\frac{\partial u_v}{\partial \overline{w}_{ij}} \quad (4.9)$$

$$\frac{\partial u_v}{\partial w_i} = f_2'\left(\sum_{i=1}^{m} w_i f_1\left(\sum_{j=1}^{n}\overline{w}_{ij}e_{vj}\right)\right).f_1\left(\sum_{j=1}^{n}\overline{w}_{ij}e_{vj}\right) \quad (4.10)$$

$$\frac{\partial u_v}{\partial \overline{w}_{ij}} = f_2'\left(\sum_{i=1}^{m} w_i f_1\left(\sum_{j=1}^{n}\overline{w}_{ij}e_{vj}\right)\right).f_1'\left(\sum_{j=1}^{n}\overline{w}_{ij}e_{vj}\right).w_i e_{vj} \quad (4.11)$$

The update rules are:

$$w_i(k) = w_i(k-1) + \Delta w_i(k) \quad (4.12)$$

$$\overline{w}_{ij}(k) = \overline{w}_{ij}(k-1) + \Delta\overline{w}_{ij}(k) \quad (4.13)$$

Similar updating rules can be applied for NNC_w controller. The training procedure for two neural network controllers NNC_v and NNC_w is summarized in the following:

Step 1: Set small random values to all weights of two neural controllers NNC_v and NNC_w

Step2: Get actual velocities from two encoders to calculate two control signals u_v and u_ω by using equation (4.6) for two sub-systems.

Step 3: Update the weights of the two neural network controllers by equations from (4.8) to (4.13), where the plant Jacobian known as the third term in (4.8) and (4.9) is calculated directly from the nominal wheelchair model (4.5).

Step 4: Repeat step 2 to step 3 until optimal weights of two neural network controllers are obtained.

Step 5: Two trained neural controllers are then used as two controllers in optimal path following scheme in Fig 1.

V. REALTIME EXPERIMENTAL RESULTS AND DISCUSSIONS

The algorithms described in the previous sections were implemented in ANSI C LabWindow CVI 8.5. The sampling time is selected as: T_s = 20 ms. A multilayer feed-forward with ***n*** input nodes, ***m*** hidden nodes and ***o*** output nodes is described as *{n,m,o}*. The final structure of NNC_v and NNC_w were chosen as *{3,4,1}* and *{3,5,1}*, respectively. The learning rates for both neural network controllers were 0.015. Convert the two element $P_D(1,1)$ and $P_D(2,2)$ into discrete form with sampling time as 20 [ms], we obtain as follows:

$$v(t) = 0.0015u_v(t) + 0.0014u_v(t-1) + 1.89v(t-1) - 0.89v(t-2)$$
$$\omega(t) = 0.023u_\omega(t) + 0.022u_\omega(t-1) + 1.87\omega(t-1) - 0.86\omega(t-2)$$

Therefore, the plant Jacobian matrix is calculated as:

$$J(t) = \begin{bmatrix} \dfrac{\partial v(t)}{\partial u_v(t)} & 0 \\ 0 & \dfrac{\partial \omega(t)}{\partial u_\omega(t)} \end{bmatrix} = \begin{bmatrix} 0.0015 & 0 \\ 0 & 0.023 \end{bmatrix}$$

After training, two neural controller weights converge to optimal values. These trained neural controllers were then used as robust controllers in the path-following control scheme in Figure 1. Position control rule was computed via (3.5) with the coefficient as: $K_x = 1; K_y = 3; K_\theta = 2$).

Two real-time experiments were conducted. Both PID controllers (designed by using Root Locus technique [8]) and the proposed robust neural controllers are implemented. In the first experiment, the wheelchair tracked a 10mx10 m square

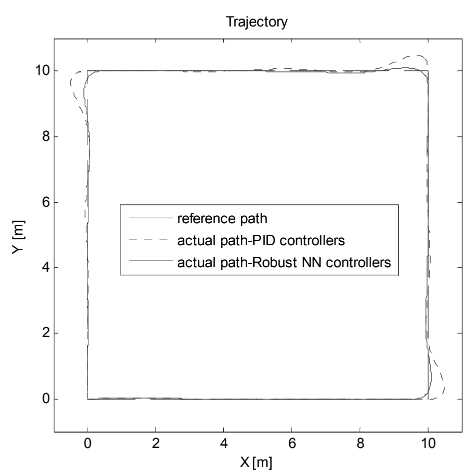

Figure 4: A 10x10 m square tracking control

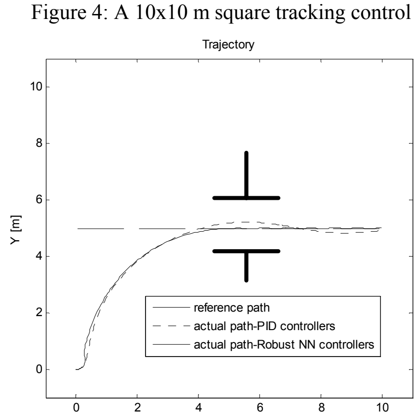

Figure 5: Door passing task

The results shown in Figure 4 prove that the tracking performance of neural controllers is better than that of PID controllers.

In the second experiment, a door passing task was used as an alternative performance assessment. The starting position of the wheelchair was the origin point of the Cartesian coordinate. The wheelchair had to track the reference path in Fig 5. A better tracking performance is achieved using two neural controllers.

VI. CONCLUSION

This paper has presented an optimal path-following control approach. In the supervisory control level, a position control rule to guarantee error position converge toward zero is proved by the use of a Lyapunov function [7]. In the drive control level, the multivariable wheelchair system is decoupled completely into two subsystems by using a systematic decoupling technique. Two neural network controllers are then designed independently to ensure robust performance of the system even under model uncertainties. A calibration procedure is employed in this wheelchair system to improve the position measurement accuracy significantly. Real-time experiments of square tracking and door passing, both confirm that robust performance has been achieved.

REFERENCES

[1] U. Borgolte, H. Hoyer, C. Bühler, H. Heck, and R. Hoelper, "Architectural Concepts of a Semi-autonomous Wheelchair," *Journal of Intelligent and Robotic Systems*, vol. 22, pp. 233 - 253, 1998.

[2] B. Lohmann, "The Autonomous Wheelchair "Easy" and its Hierarchical Control," *Mathematical and Computer Modelling of Dynamical Systems*, vol. 10, pp. 169-185, 2004.

[3] H. N. Chow and Y. Xu, "Learning Human Navigational Skill for Smart Wheelchair in a Static Cluttered Route," *IEEE Transactions on Industrial Electronics*, vol. 53, pp. 1350 - 1361, 2006.

[4] G. Bourhis and M. Sahnoun, "Assisted Control Mode for a Smart Wheelchair," *10th IEEE International Conference on Rehabilitation Robotics*, pp. 158 - 163, 2007.

[5] G. Bourhis, O. Horn, O. Habert, and A. Pruski, "An autonomous vehicle for people with motor disabilities," *IEEE Robotics & Automation Magazine* vol. 8, pp. 20 - 28, 2001.

[6] F. Espinosa, E. Lopez, R. Mateos, M. Mazo, and R. Garcia, "Application of advanced digital control techniques to the drive and trajectory tracking system of a wheelchair for the disabled," *IEEE Inter Conf on Emerging Technologies and Factory Automation*, vol. 1, pp. 521-528, 1999.

[7] Y. Kanayama, Y. Kimura, F. Miyazaki, and T. Noguchi, "A stable tracking control method for an autonomous mobile robot," *IEEE International Conference on Robotics and Automation*, vol. 1, pp. 384-389, 1990.

[8] N. T. Nguyen, H. T. Nguyen, and S. Su, "Advanced robust tracking control of a powered wheelchair system," *29th Annual International Conference of the IEEE in EMBS*, pp. 4767 - 4770, 2007.

[9] N. Hung and B. Anderson, "Triangularization technique for the design of multivariable control systems," *IEEE Transactions on Automatic Control*, vol. 24, pp. 455- 460, 1979.

[10] J. Borenstein and L. Feng, "Measurement and correction of systematic odometry errors in mobile robots," *IEEE Transactions on Robotics and Automation* vol. 12, pp. 869 - 880, 1996.