Neuro-Sliding Mode Multivariable Control of a Powered Wheelchair

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

Abstract—This paper proposes a neuro-sliding mode multivariable control approach for the control of a powered wheelchair system. In the first stage, a systematic decoupling technique is applied to the wheelchair system in order to reduce the multivariable control problem into two independent scalar control problems. Then two Neuro-Sliding Mode Controllers (NSMCs) are designed for these independent sub-systems to guarantee system robustness under model uncertainties and unknown external disturbances. Both off-line and on-line trainings are involved in the second stage. Real-time experimental results confirm that robust performance for this multivariable wheelchair control system under model uncertainties and unknown external disturbances can indeed be achieved.

I. INTRODUCTION

A smart powered wheelchair system consists of two control levels known as supervisory control level and drive control level. The supervisory level generates control commands based on the information from sensory systems and user intention. These commands are sent to the drive control level, which drives the wheelchair's actuators directly, to perform smart tasks such as obstacle avoidance and path following [1].

Various advanced control strategies for the driver control level have been developed including digital control technique [2] and neural control technique [3]. However, none of these techniques have treated powered wheelchairs as a multivariable system. As a consequence, the above control strategies would not be optimal when the operating environment for the wheelchairs is rough and nonlinear.

In term of robustness, $\underline{S}$liding $\underline{M}$ode $\underline{C}$ontrol (SMC) has been proven to be insensitive to parameter changes and external disturbances. However, it has two disadvantages: chattering phenomenon and requirement of a an exact model, which is used to calculate an equivalent control and a corrective control [4]. To solve these problems, online neuro-sliding mode control method presented in [5] uses neural networks based on sliding mode control to estimate both equivalent control and corrective control for nonlinear systems. Nevertheless, updating laws for this corrective

control are still complicated in this particular method.

This paper extends the advanced robust control technique presented in our previous work [6] by using improved neuro-sliding mode control method. The technique developed in this study includes two stages. In the first stage, a systematic decoupling technique [7] is utilised to decompose the wheelchair into two independent scalar systems. In the second stage, two NSMCs are designed for these two independent scalar systems to guarantee the robustness under external disturbances and model uncertainties. Two steps are adopted for the design of these NSMCs to achieve fast dynamic responses. At first, off-line trainings are conducted for two NSMCs to obtain nominal weights. Then, these weights are used as initial weights in on-line training for dealing with unknown external disturbances.

The paper is organized as follows. In the next section, neuro-sliding mode control design is presented. An advanced decoupling technique using neuro-sliding mode control method is described in Section III. Real-time experimental results and discussions are shown in Section IV. The conclusion is given in the section V.

II. NEURO-SLIDING MODE CONTROL DESIGN

A. Sliding mode control design

Consider an uncertainty system with external unknown disturbances in the following form:

$$\begin{cases} \dot{x} = (A + \Delta A)x + (B + \Delta B)u + \Theta d(t) \\ y = Cx \end{cases} \quad (2.1)$$

where $x \in R^n$ is system state vector, $u \in R$ is control input $A \in R^{nxn}$ is system matrix, $B \in R^{nx1}$ is input matrix and $C \in R^{1xn}$. $\Delta A \in R^{nxn}$ and $\Delta B \in R^{nx1}$ present uncertainties, $d(t)$ is external disturbance and $\Theta \in R^{nx1}$.

The sliding surface is defined as:

$$s = h^T e = h^T (x_d - x) \quad (2.2)$$

where x_d is desired state vector and $h = \begin{pmatrix} h_1 & ... & h_n \end{pmatrix}^T$ is given for s=0. From equation (2.2), the dynamic equation is cast into the error equation as follows:

$$\dot{e} = \dot{x}_d - (A + \Delta A)x - (B + \Delta B)u - \Theta d(t) \quad (2.3)$$

Let Lyapunov candidate function be selected as below:

$$V = 0.5 \cdot s^2 \quad (2.4)$$

If the derivative of the Lyapunov candidate function is chosen as follows:

$$\dot{V} = -s \cdot \delta \cdot sign(s) \qquad (2.5)$$

where δ is positive scalar and *sign(.)* is defined in (2.11a), then according to Lyaponov theorem, the dynamic system is global stable.

Taking derivative of (2.4) and equating it to (2.5), the following equation is obtained:

$$s \cdot \dot{s} = -s \cdot \delta \cdot sign(s) \qquad (2.6)$$

The time derivative of s can be obtained from (2.1) and (2.2) is given below:

$$\dot{s} = h^T \dot{x}_d - h^T (A + \Delta A)x - h^T (B + \Delta B)u - h^T \Theta d(t) \ (2.7)$$

From (2.6) and (2.7), the control input signal can be obtained as:

$$u(t) = u_{eq}(t) + u_c(t) \qquad (2.8)$$

where $u_{eq}(t)$ is the equivalent control given by:

$$u_{eq}(t) = \left[h^T (B + \Delta B)\right]^{-1} \left(h^T [\dot{x}_d - \Theta d(t)] - \left[h^T (A + \Delta A)\right]x\right) \ (2.9)$$

and $u_c(t)$ is the corrective control given by:

$$u_c(t) = \left[h^T (B + \Delta B)\right]^{-1} \cdot \delta \cdot sign(s) = K \cdot sign(s) \quad (2.10)$$

Because of the *sign(.)* function, the corrective controller in (2.10) exhibits high frequency oscillations known as chattering phenomena. A shifted sigmoid function, *g(.)*, described in (2.11b), is used to eliminate this effect.

$$sign(s) = \begin{cases} +1 & if \ s \ge 0 \\ -1 & if \ s < 0 \end{cases} \qquad (2.11a)$$

$$g(s) = \frac{2}{1 + e^{-s}} - 1 \qquad (2.11b)$$

B. Neuro-sliding mode control design

Due to uncertainties and unknown external disturbance, equivalent control in (2.9) and corrective control in (2.10) may not be directly calculated. Two NSMCs employed to estimate two control signals are described in the Figure 2.1. The equivalent control is computed by the Equivalent Neural Network (ENN). Meanwhile, the corrective control is computed by the Corrective Neural Network (CNN). Since the ENN keeps the plant states on the sliding surface s=0, the CNN is trained to drive the plants state back to sliding surface if they are out of the surface.

Both ENN and CNN are chosen as a three-layer feed-forward neural network shown in Fig 2.2 and Fig 2.3. Estimated equivalent control and estimated corrective control can be calculated as in (2.12) and (2.13) as follows:

$$\hat{u}_{eq} = K_u \cdot g\left(\sum_{j=1}^{m} W_j \cdot g\left(\sum_{i=1}^{n} \overline{W}_{i,j} \cdot Z_i\right)\right) \qquad (2.12)$$

$$\hat{u}_c = K \cdot g\left(\sum_{i=1}^{n} h_i \cdot e_i\right) \qquad (2.13)$$

where $g(.)$ is defined in (2.11) and the neural network parameters are described in Figure 2.2 and Figure 2.3.
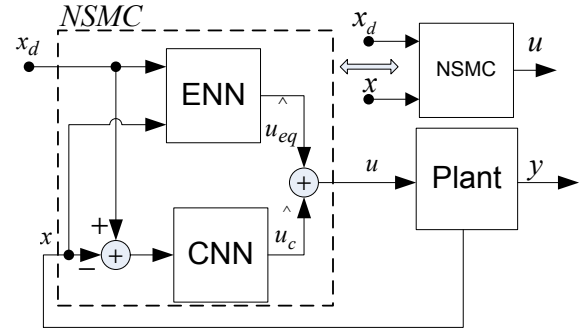

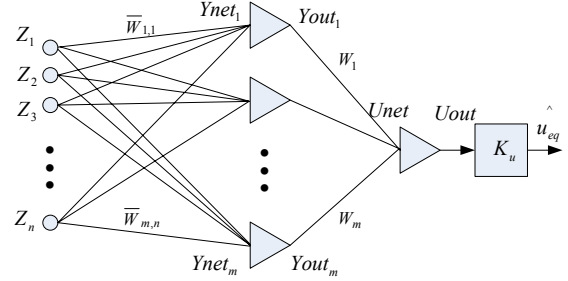Figure 2.1: The control structure

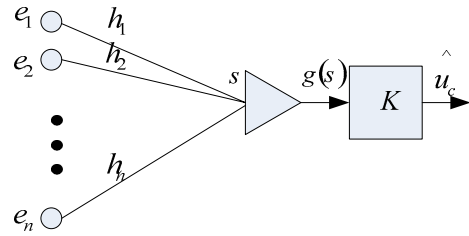Figure 2.2: The equivalent neural network structure

Figure 2.3: The corrective neural network structure

To estimate the value of the equivalent control and the corrective control, the training processes use iterative steepest descent algorithm to minimize the mean square errors between actual values and desired values, defined in (2.14).

$$E = \frac{1}{2}\left(u_{eq} - \hat{u}_{eq}\right)^2 ; J = \frac{1}{2}s^2 \qquad (2.14)$$

The weight adaptation laws for the ENN aimed at minimizing E are in following equations:

$$\Delta W_j = -\eta \frac{\partial E}{\partial W_j} = \eta\left(u_{eq} - \hat{u}_{eq}\right) \cdot k_u \frac{1}{2}\left(1 - g(Unet)^2\right) Yout_j \ (2.15)$$

In (2.15), the equivalent control u_{eq} is unknown, but the characteristic of $u_{eq} - \hat{u}_{eq}$ and the sliding surface s is similar [5]. Thus, the sliding value s can be used instead of using $u_{eq} - \hat{u}_{eq}$.

The equation (2.15) can be rewritten as follows:

$$\Delta W_j = -\eta \frac{\partial E}{\partial W_j} = \eta \cdot s \cdot k_u \frac{1}{2}\left(1 - g(Unet)^2\right) Yout_j \qquad (2.16)$$

Similarly, the weights between input-layer and hidden-layer of the ENN can be updated as follows:

$$\Delta \overline{W}_{i,j} = -\eta \frac{\partial E}{\partial \overline{W}_{i,j}} = \frac{1}{4}\eta \cdot s \cdot k_u\left(1 - g(Unet)^2\right) W_j\left(1 - g(Ynet_j)^2\right) Z_i \ (2.17)$$

The weight updating laws for the CNN to minimize the cost function J can be written in equation (2.18).

$$\Delta h_i = -\mu \frac{\partial J}{\partial h_i} = -\mu \cdot s \cdot e_i \tag{2.18}$$

where $\eta > 0$ and $\mu > 0$ are learning rate.

Assume that

$$sign(B + \Delta B) \cdot sign(B) > 0 \tag{2.19}$$

Because vector h is changed online so that the cost function J defined in (2.14) converges to zero, K in (2,10) can be calculated from the nominal model as:

$$K = \frac{\delta}{h^T B} \tag{2.20}$$

III. ADVANCED DECOUPLING CONTROL OF THE WHEELCHAIR

A. Decoupling design of the wheelchair system

To simplify the control design, decoupling technique is used to cast a multivariable problem into scalar problems. In [6], the triangularization technique is used to construct the desired decoupler D(s) for the wheelchair nominal model $G_0(s)$ as in the equation (3.2):

$$G_0(s) = \begin{bmatrix} \dfrac{1.4}{(1+0.8s)(1+0.225s)} & \dfrac{0.125}{(1+0.4s)(1+0.15s)} \\ \dfrac{0.1}{(1+0.35s)(1+0.2s)} & \dfrac{1.8}{(1+0.5s)(1+0.2s)} \end{bmatrix} \tag{3.1}$$

$$D(s) = \begin{bmatrix} 1 & -0.0893\dfrac{(0.8s+1)(0.225s+1)}{(0.4s+1)(0.15s+1)} \\ 0 & 1 \end{bmatrix} \tag{3.2}$$

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

$$P_0(s) = G_0(s)D(s) = \begin{bmatrix} \dfrac{280}{(5+4s)(40+9s)} & 0 \\ \dfrac{10}{(20+7s)(5+s)} & \dfrac{123.32(s+2.89)}{(20+7s)(5+s)(2+s)} \end{bmatrix} \tag{3.3}$$

Now, we try to diagonalize the triangular matrix $P(s)$ by choosing pre-compensator V(s) as follows:

$$V(s) = \begin{bmatrix} 1 & 0 \\ c & 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ -8.11\dfrac{2+s}{100s+289} & 1 \end{bmatrix} \tag{3.4}$$

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

$$P_D(s) = P_0(s) \cdot V(s) \approx \begin{bmatrix} \dfrac{280}{(5+4s)(40+9s)} & 0 \\ 0 & \dfrac{123.32}{(5+s)(2+s)} \end{bmatrix} \tag{3.5}$$

B. Neuro-Sliding Mode Control design of the wheelchair system

After being decoupled, the wheelchair is decomposed into two scalar systems required to design two NSMCs (Figure 3.1). Two control design steps needed for each independent subsystem are described the Figure 3.2.
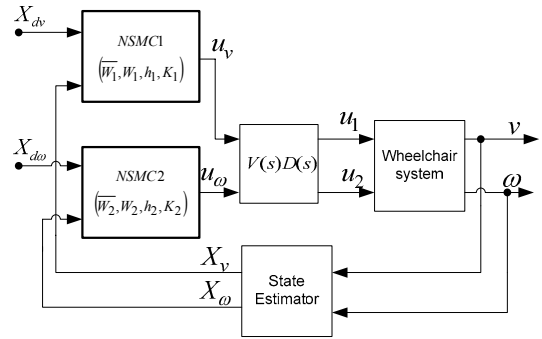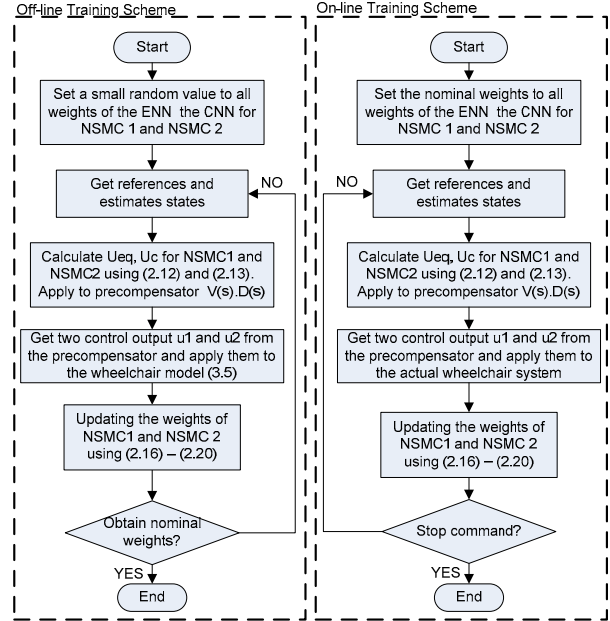

Figure 3.1: Advanced decoupling control structure

Figure 3.2: The flow chart for two kinds of training schemes.

Step1: Off-line training scheme: The diagonal model in (3.5) is used to train two corresponding NSMCs so that their weights converge to nominal values, implies that control design specifications are satisfied.

Step2: On-line training scheme: The obtained weights are then assigned to two NSMCs to reject uncertainties and external disturbances of the whole system.

IV. 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.

Define a multilayer feed-forward with *n* input nodes, *m* hidden nodes and *o* output nodes as *{n,m,o}*. The ENN and CNN structure of the NSMC 1 are *{2,3,1}* and *{2,1,1}* while that of the NSMC 2 are *{2,3,1}* and *{2,1,1}*.

The learning rates of NSMC 1 are chosen as $\eta = 0.8; \mu = 0.05$ while the learning rates of NSMC 2 are chosen as $\eta = 0.65; \mu = 0.03$. The value K_1 and K_2 are directly calculated from (3.4) with $\delta = 0.035$. $K_u = 1$ for both NSMCs.

Results:

At the beginning, off-line training scheme is executed for the decoupled two scalar systems (see equation (3.5)) to get nominal weights of the two neuro-sliding mode controllers NSMC 1 and NSMC 2.

As a result, the nominal weights of the NSMC 1 for linear velocity loop are obtained as follows:

$$\overline{W}_1 = \begin{bmatrix} -4.62689 & -1.79929 \\ -2.35282 & 7.17887 \\ 0.64789 & 7.43743 \end{bmatrix}; W_1 = \begin{bmatrix} 0.40276 \\ -1.88016 \\ -6.32161 \end{bmatrix}; h_1 = \begin{bmatrix} 0.6441 \\ 1.1769 \end{bmatrix}; K_1 = 1.38$$

The nominal weights of the NSMC 2 for angular velocity loop are:

$$\overline{W}_2 = \begin{bmatrix} -2.08754 & 5.15314 \\ -1.10662 & 4.39517 \\ 1.64000 & -5.88757 \end{bmatrix}; W_2 = \begin{bmatrix} -4.95932 \\ -6.38345 \\ 0.06663 \end{bmatrix}; h_2 = \begin{bmatrix} 0.6119 \\ 1.2066 \end{bmatrix}; K_2 = 1.42$$

Then, online-training scheme with trained NSMCs is implemented and applied in two real-time experiments.

As a result, in the first experiment, the wheelchair tracks a circle with radius 1 m (v=0.377 m/s; ω =0.38 rad/s). Figure 4.1 shows the reference and actual experimental trajectories and the closed-loop responses of two decoupled loops.
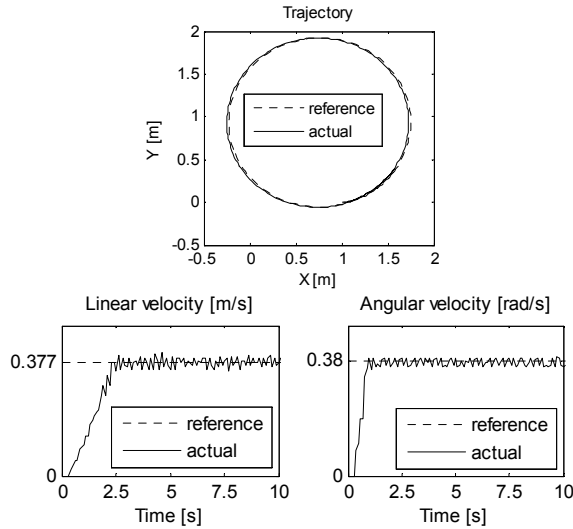

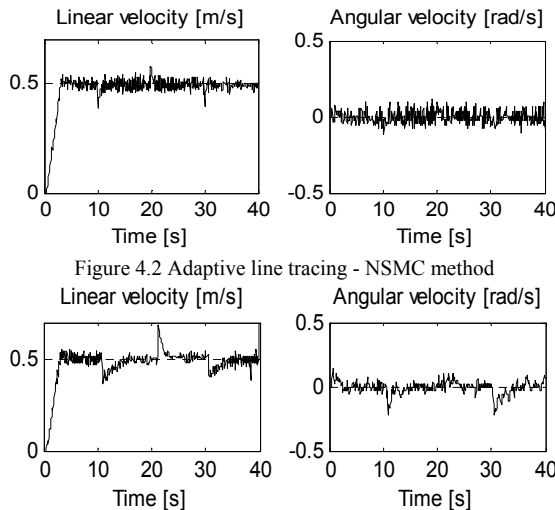Figure 4.1 Circular tracing task.

Figure 4.2 Adaptive line tracing - NSMC method

Figure 4.3 Adaptive line tracing - old method

In the second experiment, we test robust tracking performance of the designed wheelchair system under external disturbance for a line tracing task. Specifically, a person weighted 60 kg artificially generates external disturbances by pulling the wheelchair at t=10s, releasing it at t=20s, and pulling it again at t=30s. In this experiment, the wheelchair runs on the wooden surface at first at t=20s, then it runs on the carpet.

Figure 4.2 shows the closed-loop responses of linear velocity and angular velocity. A comparison study is also performed by repeating the experiment for the controller which we proposed in [6]. Figure 4.3 is the responses with the use of the control method in [6].

Discussions

Figure 4.1 shows the closed-loop responses satisfy control design requirements with settling time $T_s \leq 2.5$ s and no overshoot. It also demonstrates that the two velocity loops can be controlled independently by the two NSMCs without coupling effects. Compared to Figure 4.3, Figure 4.2 shows that NSMC achieves a significant improvement in the sense of how it can substantially suppress external disturbances by using online learning algorithm.

V. CONCLUSION

In this paper, we have come to extend our advanced robust technique in [6] for the control of a wheelchair system. First, the multivariable system is decoupled completely into two scalar subsystems by using a pre-compensator. Two NSMCs are then designed independently for the two subsystems based on the combination of neural network design and sliding mode theory. This good combined method can guarantee the robustness of the closed loop system under uncertainties and unknown external disturbances. Real-time experiment results confirm that robust performance for the multivariable wheelchair control system has been achieved.

REFERENCES

[1] G. Bourhis and Y. Agostini, "Man-machine Cooperation for the Control of an Intelligent Powered Wheelchair," *Journal of Intelligent and Robotic Systems*, vol. 22, pp. 269 - 287, 1998.

[2] L. Boquete, R. Barea, R. Garca, M. Mazo, and F. Espinosa, "Identification and control of a wheelchair using recurrent neural networks " *Engineering Applications of Artificial Intelligence* vol. 12, pp. 443-452, 1999.

[3] R.-X. Chen, L.-G. Chen, and L. Chen, "System Design Consideration for Digital Wheelchair controller," *IEEE Trans on Industrial Electronics*, vol. 47, pp. 898-907, 2000.

[4] J. Y. Hung, W. Gao, and J. C. Hung, "Variable structure control: a survey," *IEEE Transactions on Industrial Electronics*, vol. 40, pp. 2 - 22, 1993.

[5] M. Ertugrula and O. Kaynak, "Neuro sliding mode control of robotic manipulators," *Mechatronics* vol. 10, pp. 239-263, 2000.

[6] 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.

[7] 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.