2003 International Symposium on Advanced Intelligent Systems
September 25-28, 2003, Jeju, Korea

Neural Network Based Disturbance Canceler with Feedback Error Learning for Nonholonomic Mobile Robots

Kiyotaka Izumi, Rafiuddin Syam, Keigo Watanabe, and Kazuo Kiguchi
Department of Advanced Systems Control Engineering, Graduate School of Science and Engineering,
Saga University, 1 Honjomachi, Saga 840-8502, Japan
E-mail: izumi@saga-u.ac.jp

***Abstract*— Conventional disturbance rejection methods have to derive the inverse model of a system. However, the inverse model of a nonholonomic system is not unique, because an inverse it changes depending on initial conditions and desired values. A kind of internal model control (IMC) using feedback error learning is discussed for the motion control of nonholonomic mobile robots in this paper. The present method is different from a conventional IMC whose control system consists of an inverse model, a direct model and a filter. The present disturbance rejection method need not use a direct model, where the remaining two elements are composed of the same inverse model based on neural networks.**

I. Introduction

It is the feature of nonholonomic systems, that the number of control inputs is less then the degree of freedom of the system. The conventional control approaches to nonholonomic systems are exact linearization, chained form method, back-stepping method[1], and neural networks approach[2]. In the control of the driving wheel type mobile robot with nonholonomic constraint, Fukao et al. [1] combines a kinematic controller and a torque controller using back-stepping approach. Fierro[2] recommends the use of a velocity tracking inner loop, and proposes the online compensator with neural networks.

On the other hand, the internal model control (IMC) was proposed to cancel the disturbance for nonlinear systems[3], [4], [5]. The structure of the IMC is simple. Components of the IMC are the internal model, the filter, and the controller. Theoretically, if we obtain the accurate inverse model of the system, then the disturbance is canceled wholly using the IMC. However, we can't obtain the unique inverse model of the nonholonomic systems.

One of attractive approaches is the neuro interface[6], in which the learning mechanism of neural networks consists of the specialized learning. This method shows efficient results with application for the trailer control. This control system is constructed with the controller, the reference model and the filter. The structure of this method is quite similar to IMC. Unfortunately, any relation between neuro interface and IMC is not discussed in the paper[6].

In this paper, we propose a neuro interface-like control system using the concept of IMC. Nonholonomic system does not have an unique inverse model. However the present method need not use the accurate inverse model. The neural network is here trained by the feedback error learning. The effectiveness of the proposed method is shown by applying it for the mobile robot with two independent driving wheel.

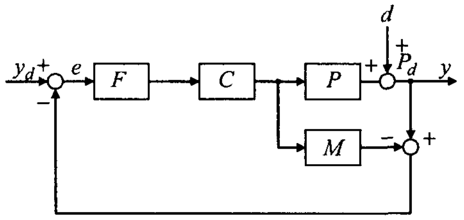

Fig. 1. Block diagram of IMC

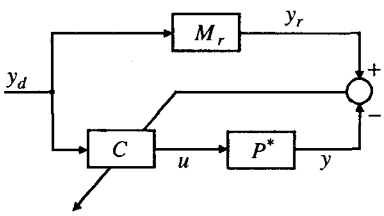

Fig. 2. Training of an inverse model of neuro interface

II. Internal Model Control

If the nonlinear system is stable, the IMC is the effective control method. The IMC concept is illustrated in **Fig. 1**, where y_d is the desired value, y is the output of the system, e is the error between y_d and y, and d is the disturbance. F is the filter, C is the controller, P is the plant, and M is the model of the plant. The advantages of the IMC are as follows:

- If the plant including C and d is stable and we can use M, then the closed-loop system becomes input-output stable.
- If C is equivalent with P^{-1}, then it is compensated such that y becomes y_d.
- If C is equivalent with M^{-1}, then the closed-loop system becomes input-output stable.

In order to apply the above features, it is very important to obtain an inverse model M^{-1} that is input-output stable and to acquire the model $M \simeq P$.

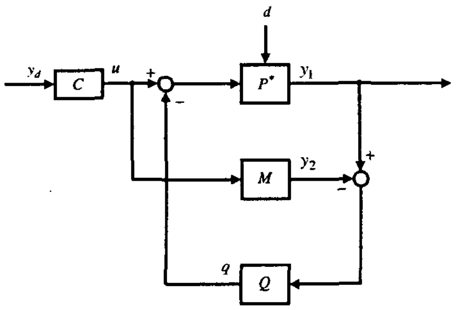

Fig. 3. Neuro interface and disturbance canceler

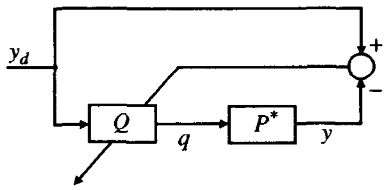

Fig. 4. Training of a filter Q

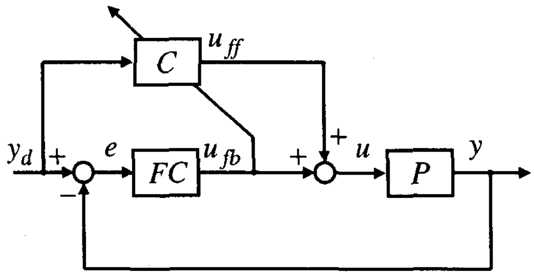

Fig. 5. Training of an inverse model through feedback error learning

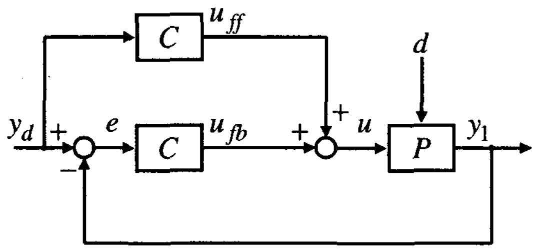

Fig. 6. Block diagram of disturbance cancel control using feedback error learning architecture

III. Neuro Interface

Neuro interface is one of control approaches using neural networks for nonlinear systems[6]. The approach is developed for interface system of complex and nonlinear systems. The neuro interface is trained by **Fig. 2**, where M_r is the reference model, C is the neuro controller, P^* is the stabilized plant, y_d is the desired value, y_r is the output of M_r and y is the output of the stabilized plant. C is trained by the specialized learning method, in which the teaching signal is y_r. Therefore, we obtain the following relationships:

$$\begin{aligned} y_r &= M_r y_d & (1)\\ y &= P^* u & (2)\\ &= P^* C y_d & (3)\\ &= P^* C M_r^{-1} y_r. & (4) \end{aligned}$$

The controller has to become

$$C = P^{*-1} M_r \qquad (5)$$

to assure $y_r = y$. Thus, the neuro mapping should be acquired as the product between the inverse plant P^{*-1} and the reference model M_r.

The disturbance cancel system by neuro interface system using the trained controller is illustrated by **Fig. 3**, where Q is the filter. Q has to be trained by specialized learning illustrated in **Fig. 4**, where the teaching signal is y_d. In theory, Q has to become P^{*-1}. If the plant is a linear system, the neuro interface becomes the optimal disturbance cancel system. Unfortunately, the optimality of the neuro interface is not assured in the nonlinear system. Nevertheless, the neuro interface is shown to be effective in the nonlinear system from simulation results[6].

IV. Disturbance Canceler with Feedback Error Learning

We propose a disturbance canceler system using the feedback error leaning for the nonholonomic system. The concept is similar to IMC, but we extend the concept of IMC to apply for the nonholonomic system. The proposed system is trained by the feedback error learning illustrated in **Fig. 5**. To stabilize the plant P, any approach is acceptable for the feedback controller FC. In the sequel, the control system is reduced to one shown in **Fig. 6** using the trained C in Fig. 5. In the present method, there is no internal model, which is slightly different from IMC. In fact, the structure consists of the filter and the controller.

From Fig. 6, we obtain

$$\begin{aligned} y &= P(u_{ff} + u_{fb}) & (6)\\ y_1 &= y + \Delta y(d) & (7)\\ u_{fb} &= Ce &\\ &= C(y_d - y - \Delta y(d)) &\\ &= C(y_d - Pu_{ff} - Pu_{fb} - \Delta y(d)) & (8) \end{aligned}$$

where $\Delta y(d)$ is the output error due to any disturbance. Then, the relationship between $\Delta y(d)$ and u_{fb} is reduced to

$$u_{fb} = -\frac{1}{2} P^{-1} \Delta y(d). \qquad (9)$$

From this equation, if P^{-1} is stable, then u_{fb} is expected to be zero, independing on the accuracy of the inverse model.

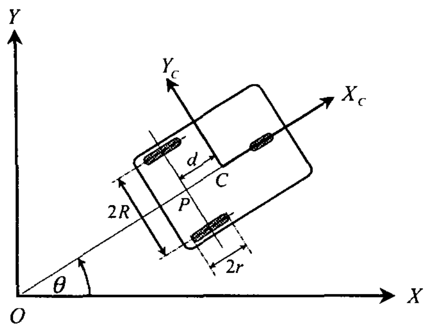

Fig. 7. Nonholonomic mobile robot

TABLE I
PARAMETERS OF ROBOT IN SIMULATIONS

m [kg]	I [kgm^2]	R [m]	r [m]	d [m]
10	5	0.5	0.05	0.2

V. SIMULATIONS

A. Mobile robot

A nonholonomic mobile robot is illustrated by **Fig. 7**. The robot can be described by

$$M(q)\ddot{q} + V(q,\dot{q})\dot{q} = B(q)\tau - A^T(q)\lambda \tag{10}$$

where q is the generalized coordinates, $q = [x_c\ y_c\ \theta]$, $(x_c\ y_c)$ is the robot position of the center of gravity, θ is the azimuth, M is an inertia matrix, V is the centrifugal and coriolis matrix, B is the input distribution matrix, A is the matrix associated with the constrains, and λ is the vector of constraint forces. The kinematic equality constraints can be expressed by $A(q)\dot{q} = \mathbf{0}$. The kinematic equality constraint of the mobile robot is

$$\dot{y}_c \cos\theta - \dot{x}_c \sin\theta - d\dot{\theta} = 0. \tag{11}$$

Finally, matrix or vector elements of (10) are derived as follows:

$$\begin{aligned}
M(q) &= \begin{bmatrix} m & 0 & md\sin\theta \\ 0 & m & -md\cos\theta \\ md\sin\theta & -md\cos\theta & I \end{bmatrix} \\
V(q,\dot{q}) &= \begin{bmatrix} 0 & 0 & md\dot{\theta}\cos\theta \\ 0 & 0 & md\dot{\theta}\sin\theta \\ 0 & 0 & 0 \end{bmatrix} \\
B(q) &= \frac{1}{r}\begin{bmatrix} \cos\theta & \cos\theta \\ \sin\theta & \sin\theta \\ R & -R \end{bmatrix} \\
A(q) &= [-\sin\theta\ \cos\theta\ -d] \\
\lambda &= -m(\dot{x}_c\cos\theta + \dot{y}_c\sin\theta)\dot{\theta}
\end{aligned}$$

where m denotes the mass, I denotes the inertia, d denotes the distance between the driving axis and the center of mass, $2R$ denotes the distance of wheels, and r denotes the radius of wheel. The physical parameters are shown in **Table I**.

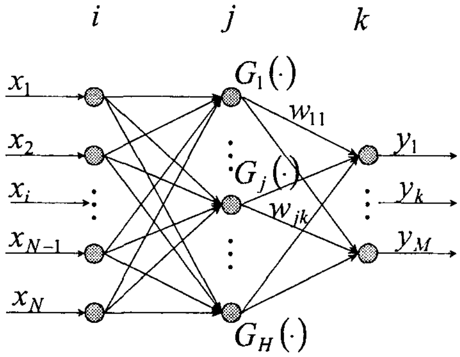

Fig. 8. Structure of RBFNN

TABLE II
GA PARAMETERS

Parameter size	72
Population size	60
Bit size of an individual	576
Selection	Tournament method with 3 individuals
Crossover	Uniform crossover
Crossover ratio	0.6
Mutation ratio	1/576
Stop condition	5000 generations
The number of elites	6

B. Design of feedback controller

An alternative representation of equation (10) is as follows:

$$\begin{bmatrix} m & 0 \\ 0 & I - md^2 \end{bmatrix}\begin{bmatrix} \dot{v} \\ \ddot{\theta} \end{bmatrix} = \frac{1}{r}\begin{bmatrix} 1 & 1 \\ R & -R \end{bmatrix}\begin{bmatrix} \tau_r \\ \tau_l \end{bmatrix} \tag{12}$$

where v is the forward velocity of mobile robot, τ_r is the driving torque of right wheel and τ_l is the driving torque of left wheel. When the new state variables are defined as $x = [v\ \theta\ \dot{\theta}]^T$, the state equation is given by

$$\begin{aligned}
\dot{x} &= Ax + B\tau \qquad (13) \\
A &= \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix} \\
B &= \frac{1}{r}\begin{bmatrix} 1/m & 1/m \\ 0 & 0 \\ R/(I - md^2) & -R/(I - md^2) \end{bmatrix}
\end{aligned}$$

where $\tau = [\tau_r\ \tau_l]^T$. The controller FC in Fig. 5 is assumed to be based on a pole placement approach, because the equation (13) is the linear system and controllable matrix of equation (13) is full rank. Thus, the feedback control rule is

$$\tau = -Kx \tag{14}$$

where K is the feedback gain. The designed poles are $[-2\ -4\ -6]$, in which the resultant K is as follows:

$$K = \begin{bmatrix} 0.9572 & 2.2260 & 1.6467 \\ 1.0734 & -3.2329 & -2.0052 \end{bmatrix}.$$

C. RBFNN

The present disturbance canceler is based on an RBFNN illustrated in **Fig. 8**. The jth unit function of the hidden layer is

$$G_j(\boldsymbol{x}) = \exp\left(-\frac{\|\boldsymbol{x} - \boldsymbol{c}_j\|^2}{\sigma^2}\right). \quad (15)$$

The unit function of the output layer is sigmoid type:

$$y_k = \frac{1}{1+\exp(z_k)} - 0.5 \quad (16)$$

$$z_k = \sum_{j=1}^{H} w_{jk} G_j(\boldsymbol{x})$$

where H is the number of hidden layer's units. For the training of C, inputs of RBFNN are reference vector $[x_{cr}\ y_{cr}\ \theta_r]$, and output is $\boldsymbol{u}_{ff}$. In the simulation, H is set to 20.

The training of RBFNN parameters was carried out using the genetic algorithm (GA). Parameters of GA operations are shown in **Table II**. The cost function of GA is

$$fitness = \frac{1}{0.05}\int_0^{10} \|\boldsymbol{u}_{fb}\|^2 dt$$

where the simulation time is 10 [s] and the sampling interval is 0.05 [s].

D. Simulation results

In the training, reference trajectory is set to

$$x_{cr} = 2.5\sqrt{2} + 0.5t\cos\theta_r$$
$$y_{cr} = 2.5\sqrt{2} + 0.5t\sin\theta_r$$
$$\theta_r = -3\pi/4$$

and the initial values are set to $[x_c(0)\ y_c(0)\ \theta(0)] = [2.5\sqrt{2}\ 2.5\sqrt{2}\ -3\pi/4]$.

Figure 9 illustrates the simulation result of the RBFNN trained in Fig. 5. The robot can't track to the reference trajectory. It is found from Fig. 9 that the training of RBFNN isn't sufficient.

Although the training result is insufficient, the trained RBFNN is applied for the present disturbance canceler shown in Fig. 6. The initial value is different from the training case. The robot trajectory is illustrated in **Fig. 10**. Note that, the initial value of solid line is set to $[x_c(0)\ y_c(0)\ \theta(0)] = [2.5\sqrt{2} - 0.5\ 2.5\sqrt{2}\ -\pi/4]$, and that of dotted dash line is set to $[x_c(0)\ y_c(0)\ \theta(0)] = [2.5\sqrt{2} - 0.01\ 2.5\sqrt{2} - 0.01\ -3\pi/4 - 0.01]$. If the initial change as a disturbance closes to the training case, then the feedback control inputs become large. Thus, when the initial change is relatively small, the robot behaves straight movement. On the contrary, if the initial change becomes outside of the training range, then the outputs of RBFNN as a feedback controller have some oscillations. Therefore, the robot behaves rotational movement in the latter half of simulation. It is considered from simulation results that RBFNN as a feedback controller has to be further tuned online to obtain a better performance on the disturbance rejection.

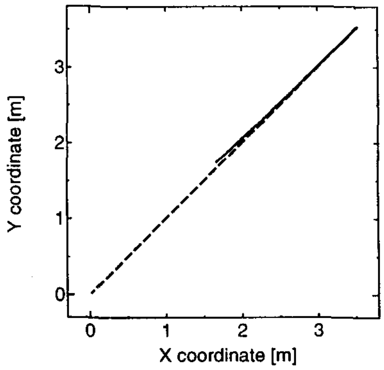

Fig. 9. Simulation result using trained parameters

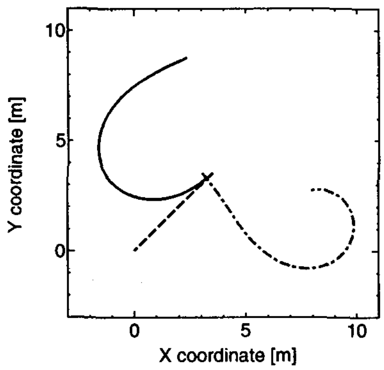

Fig. 10. Simulation result of disturbance canceler

VI. Conclusions

In this paper, we have discussed about the possibility of the neuro interface-like control system using the concept of IMC. Advantage of the proposed method is that it need not use any direct model of the plant as required by the neuro interface and the internal model control. Simulation results were insufficient. However, if parameters of the neural network could be adjusted suitably, then the present method would become effective. Furthermore, when any online tuning is added to C in the feedback loop of Fig. 6, the performance will be improved.

References

[1] T. Fukao, H. Nakagawa, and N. Adachi, "Adaptive Tracking Control of a Nonholonomic Mobile Robot," *IEEE Trans. on Robotics and Automation*, vol. 16, no. 5, pp. 609–615, 2000.

[2] R. Fierro and F. L. Lewis, "Control of a Nonholonomic Mobile Robot Using Neural Networks," *IEEE Trans. on Neural Networks*, vol. 9, no. 4, pp. 589–600, 1998.

[3] C. Kambhampati, R. J. Craddok, M. Tham, and K. Warwick, "Inverse Model Control Using Recurrent Networks," *Mathematics and Computers in Simulation*, vol. 51, pp. 181–199, 2002.

[4] D. M. Wolpert, R. C. Miall, and M. Kawato, "Internal Models in the Cerebellum," *Trends in Cognitive Sciences*, vol. 2, no. 9, pp. 338–347, 1998.

[5] D. M. Wolpert and M. Kawato, "Multiple Paired Forward and Inverse Models for Motor Control," *Neural Networks*, vol. 11, pp. 1317–1329, 1998.

[6] B. Widrow and M. M. Lamego, "Neurointerfaces," *IEEE Trans. on Control Systems Technology*, vol. 10, no. 2, pp. 221-228, 2002.