Control of Nonholonomic Mobile Robots Using a Neurointerface with a Fuzzy Feedback Compensator

Rafiuddin Syam, Keigo Watanabe, Sang-Ho Jin*, and Kiyotaka Izumi
Department of Advanced Systems Control Engineering
Saga University, 1-Honjomachi, Saga 840-8502, Japan
*Department of Mechanical Engineering, Doowon Technical College
678, Jangwon-ri, Juksan-myon, Ansung-shi, Kyonggi-do, 456-718 Korea
02td74@edu.cc.saga-u.ac.jp, {watanabe,izumi}@me.saga-u.ac.jp,
shjin@doowon.ac.kr

***Abstract*— A design method for neural network (NN) based feedforward controller in the framework of neurointerface has been proposed for nonholonomic robots by applying a concept of virtual master-slave system, in which a master robot was assumed to have a stable inverse dynamical model that includes unknown physical parameters. To reduce an output deviation caused by the mapping error of NN or a change of the mass of the robot, we here introduce a PD-based feedback compensator or an adaptive fuzzy compensator. The effectiveness of the present approach is shown by a simulation for a tracking control problem of a nonholonomic mobile robot with two-independent driving wheels.**

I. Introduction

Widrow and Lamego[1] proposed a neurointerface (WL-neurointerface) approach to ease the control of nonholonomic systems by nonexpert. This method is composed mainly of two parts: one is an inverse system realized by neural network (NN) to generate a feedforward control input according to a reference value or the output of a reference model and the other is a feedback mechanism to suppress the effect of disturbances due to the change of initial state, mapping errors of NN, etc. This approach is explored[2] to be very similar to IMC approach[3] and it can also be regarded as one method of the so-called two-degrees-of-freedom design for robust control.

Note however that Widrow and Lamego[1] suggested no systematic ways for constructing NNs for nonholonomic systems, though a usage of any tapped delay inputs was recommended for the NN, and that Izumi et al.[2] also reported a fact that a good result cannot be obtained for a nonholonomic system, even if a feedback error learning mechanism is applied forcedly to acquire an inverse dynamical system. This is attributed to a reason that a unique inverse dynamical system cannot be solved for nonholonomic systems, because most of them include an unstable zero dynamics.

The present authors have already proposed a method for constructing a feedforward part in the framework of neurointerface for a nonholonomic mobile robot by applying a concept of virtual master-slave systems[4]. It was assumed that there exists an inverse dynamics for a master robot that can be represented by a steering model and that the transformation from the generalized coordinate of a slave robot to the coordinate of a master robot was known. Simulation results for a case, where the dynamical and kinematic parameters except for the offset distance of steering axis d are all unknown, have been already reported in Syam et al.[5].

Note however that the inverse system acquired by NN will generally yield a modeling error between it and the ideal model-based inverse system obtained from a known master robot, because a finite amount of output error is normally used to terminate the training process. Therefore, to suppress the effect of such modeling errors, changes of initial configurations (initial disturbances), and unexpected disturbances for test cases, we further have to introduce any feedback controller. From this point of view, we have already proposed a neurointerface with a PD feedback controller in Syam et al. [6] to reduce the effect of mapping error. However it should be noted that such a PD controller is effective for a fixed or slowly time-varying environment such as a case where there are no sudden changes of mass of the robot.

In this paper, we further consider a case when an adaptive fuzzy compensator is introduced. In particular, we here investigate a mixing use of a neurointerface and a PD controller or a PD controller plus a fuzzy compensator as a simple feedback compensation. The effectiveness of such a method is demonstrated by simulations.

In what follows, we first review a concept of virtual master-slave system in section II. In sections III and IV, we derive a learning method for the design parameters in a simplified fuzzy reasoning to obtain an adaptive fuzzy compensator. In section V, several simulation results are presented to demonstrate the effectiveness of the present method.

II. A Concept of Virtual Master-Slave System

Let the nonlinear plant to be controlled be described by a general nonholonomic two-wheeled robot,

$$\ddot{\boldsymbol{q}}(t) = \boldsymbol{f}_S(\dot{\boldsymbol{q}}(t), \boldsymbol{q}(t), \boldsymbol{\tau}(t)) \tag{1}$$

where $\boldsymbol{q}(t) \triangleq [x(t) \quad y(t) \quad \theta(t)]^T$ is the generalized coordinate vector, in which let the center of gravity of the robot be (x, y) and the azimuth of the robot be θ. Moreover, $\boldsymbol{f}_S \in \Re^3$ and $\boldsymbol{\tau} \triangleq [\tau_r(t) \quad \tau_l(t)]^T \in \Re^2$, where τ_r and τ_l are the driving torques of the right and left wheels, respectively. This inverse

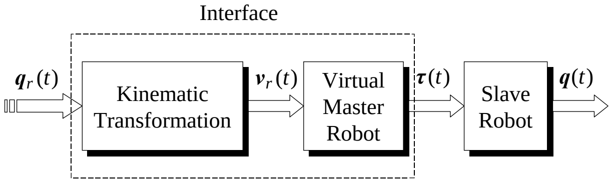

Fig. 1. Construction of an interface using master-slave concept

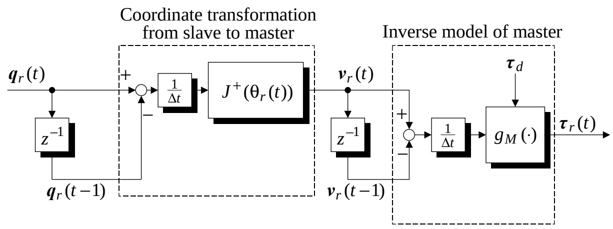

Fig. 2. Model-based feedforward controller with a virtual master-slave system

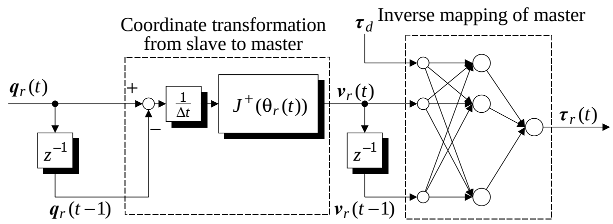

Fig. 3. NN-based feedforward controller with a virtual master-slave system

dynamical model cannot be solved stably and uniquely, so that we further consider the so-called steering model,

$$\dot{\boldsymbol{v}}(t) = \boldsymbol{f}_M(\boldsymbol{\tau}(t)) \tag{2}$$

where $\boldsymbol{v}(t) \stackrel{\triangle}{=} [v(t) \quad \dot{\theta}(t)]^T$ and $\boldsymbol{f}_M \in \Re^2$, where $v(t)$ denotes the translational velocity of the robot. The inverse of this model is known to be solved stably and uniquely.

Hereafter, it is assumed that the model (1) represents the slave robot to be controlled, while the model (2) represents a master robot to control the slave robot described in Eq. (1).

A. Torque Generation by an Inverse Model of Master Robot

In general, we can solve the inverse model of the steering model such as

$$\boldsymbol{\tau}(t) = \boldsymbol{g}_M(\dot{\boldsymbol{v}}(t)) \tag{3}$$

where $\boldsymbol{g}_M \in \Re^2$ is a stable and unique vector-valued inverse function of $\boldsymbol{f}_M$. In order to discretely give $\boldsymbol{v}(t)$ at any time t, we here consider a backward difference model approximation for $\dot{\boldsymbol{v}}(t)$. Then, the above equation can be reduced to

$$\boldsymbol{\tau}(t) = \boldsymbol{g}_M([\boldsymbol{v}(t) - \boldsymbol{v}(t-1)]/\Delta t) \tag{4}$$

where Δt is the sampling width. Given the reference velocity vectors $\boldsymbol{v}_r(t)$ and $\boldsymbol{v}_r(t-1)$ at times t and $t-1$ for the master robot, we can easily obtain the desired input torque vector $\boldsymbol{\tau}_r(t)$ at time t using the above relation.

At this stage, we must be aware of the direct kinematic relation given by

$$\dot{\boldsymbol{q}}(t) = J(\theta(t))\boldsymbol{v}(t),\; J(\theta(t)) = \begin{bmatrix} \cos(\theta) & -d\sin(\theta) \\ \sin(\theta) & d\cos(\theta) \\ 0 & 1 \end{bmatrix} \tag{5}$$

because the slave robot has its desired reference as $\boldsymbol{q}_r(t)$, where d denotes an offset distance of steering axis. Therefore, the references $\boldsymbol{v}_r(t)$ and $\boldsymbol{v}_r(t-1)$ for the master robot can be generated by

$$\boldsymbol{v}_r(t) = J^+(\theta_r(t)) \left[\frac{\boldsymbol{q}_r(t) - \boldsymbol{q}_r(t-1)}{\Delta t} \right] \tag{6}$$

$$\boldsymbol{v}_r(t-1) = J^+(\theta_r(t-1)) \left[\frac{\boldsymbol{q}_r(t-1) - \boldsymbol{q}_r(t-2)}{\Delta t} \right] \tag{7}$$

where $J^+(\cdot)$ denotes the pseudoinverse matrix of $J(\cdot)$.

Thus, given the desired references $\boldsymbol{q}_r(t)$ for the slave robot, we can discretely generate $\boldsymbol{\tau}(t)$ by using Eqs. (4) to (7). Figure 1 shows the construction of an interface using a master-slave concept.

B. Addition of a Known Torque Compensation due to Floor Frictions etc.

As can be found from Eq. (4), if $\boldsymbol{v}_r(t)$ is constant, then the desired torque $\boldsymbol{\tau}_r(t)$ will be zero in the sequel. However, in practice, there exist some disturbances such as floor frictions etc., so that we include a small disturbance torque $\boldsymbol{\tau}_d(t)$ to avoid a zero input torque such as

$$\boldsymbol{\tau}_r(t) = \boldsymbol{g}_M([\boldsymbol{v}_r(t) - \boldsymbol{v}_r(t-1)]/\Delta t, \boldsymbol{\tau}_d) \tag{8}$$

Figure 2 shows the block diagram for a feedforward controller by using the inverse dynamical model of a virtual master robot with backward difference approximation, together with a coordinate transformation with a pseudoinverse of a Jacobian matrix. This system is here called a "model-based feedforward controller" to control a slave robot as the final controlled objective.

III. Construction of Fuzzy Feedback Compensator

We have already proposed a neurointerface with a PD feedback controller as shown in Fig. 4 to reduce the effect of mapping error when constructing a feedforward controller through an inverse dynamical model of the virtual master robot. However it should be noted that such a PD controller is effective for a fixed or slowly time-varying environment such as a case where there are no sudden changes of mass of the robot.

In this paper, we further consider a case when the mass of the robot will be changed drastically, as a disturbance. In particular, we here introduce an adaptive fuzzy compensator as shown in Fig. 5 and Fig. 6.

A. Fuzzy Reasoning with a Simplified Reasoning

Let us consider a case with n input variables $(e_1, ..., e_n)$ and p output variables $(\tau_{F1}, ..., \tau_{Fp})$ as the consequent.

The simplified reasoning method can be interpreted as a special case of the Sugeno's fuzzy reasoning. In fact, this

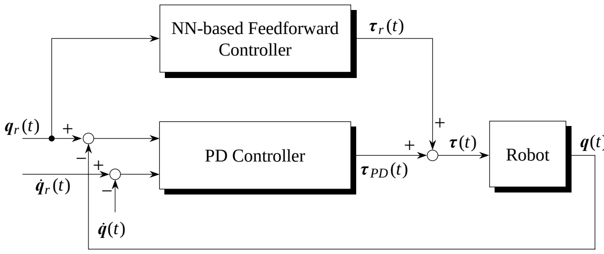

Fig. 4. A neurointerface with a PD feedback controller

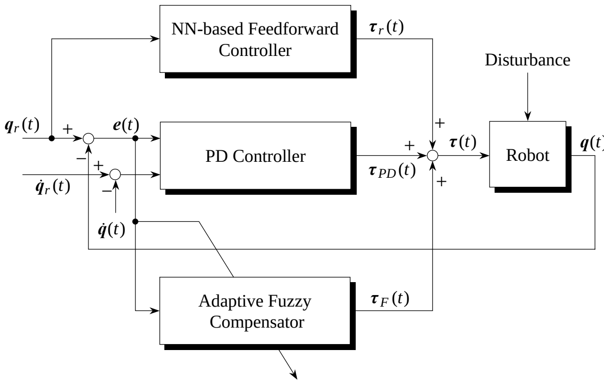

Fig. 5. A neurointerface with a PD feedback controller and an adaptive fuzzy compensator

method coincides with a case when a function in the conclusion becomes a constant w_{ij}, or a case when the width of the fuzzy set in the conclusion of the min-max-centroidal method becomes an infinitesimal value, that is, a singleton. Therefore, any i-th control rule can be written by

$$\mathrm{R}_i: \quad \text{If } e_1 = A_{i1} \text{ and ... and } e_n = A_{in} \\ \text{then } \tau_{F1} = w_{i1} \text{ and ... and } \tau_{Fp} = w_{ip} \tag{9}$$

where R_i denotes the i-th control rule, A_{ij} the fuzzy set (or fuzzy variable) in the antecedent associated with the j-th input variable at the i-th control rule, and w_{ij} denotes a constant associated with the j-th variable in the conclusion at the i-th control rule. Applying n confidences $\mu_{A_{i1}}(e_1), ..., \mu_{A_{in}}(e_n)$, the confidence in the antecedent h_i is defined by

$$h_i = \mu_{A_{i1}}(e_1) \cdot \mu_{A_{i2}}(e_2) ... \cdot \mu_{A_{in}}(e_n) \tag{10}$$

where "·" is the algebraic product. If the fuzzy set $A_{ij}(e_j)$ is adopted as a Gaussian like membership function, it follows that

$$\mu_{A_{ij}}(e_j) = \exp\{\ln(0.5)(e_j - c_{ij})^2 w_{dij}^2\} \tag{11}$$

Here, c_{ij} denotes the center value (e.g. the mean value) associated with the membership function for the j-th input data at the i-th rule, and w_{dij} denotes the reciprocal value of the deviation from the center c_{ij} to which the Gaussian function of the j-th input data at the i-th rule has value 0.5.

Then, the j-th output consequent can be calculated as the following weighted mean of $f_{ij}(\cdot)$ with respect to the weight h_i:

$$\tau_{Fj}^* = \frac{\sum_{i=1}^r h_i w_{ij}}{\sum_{i=1}^r h_i}, \quad j = 1, ..., p \tag{12}$$

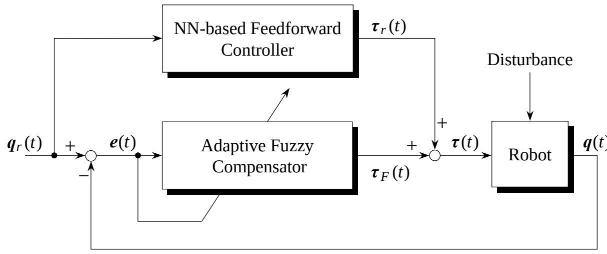

Fig. 6. A neurointerface with only an adaptive fuzzy compensator

where r denotes the total number of control rules; if the number of membership functions (i.e., the number of labels) in the antecedent is ℓ, then in general $r = \ell^n$.

B. Learning of Consequent Part

We here consider the following squared error as a cost function:

$$J = \frac{1}{2} \sum_{k=1}^{3} e_k^2(t) \tag{13}$$

where $e_k(t) = q_{rk}(t) - q_k(t)$. Then we have the update algorithm for the constant in the consequent part such as

$$w_{ij}(t+1) = w_{ij}(t) - \eta \frac{\partial J}{\partial w_{ij}(t)} \tag{14}$$

Here, it follows that

$$\frac{\partial J}{\partial w_{ij}(t)} = \sum_{k=1}^{3} \frac{\partial J}{\partial e_k(t)} \frac{\partial e_k(t)}{\partial w_{ij}(t)} \tag{15}$$

$$= \sum_{k=1}^{3} e_k(t) \frac{\partial e_k(t)}{\partial q_k(t)} \frac{\partial q_k(t)}{\partial w_{ij}(t)} \tag{16}$$

$$= -\sum_{k=1}^{3} e_k(t) \frac{\partial q_k(t)}{\partial w_{ij}(t)} \tag{17}$$

$$\frac{\partial q_k(t)}{\partial w_{ij}(t)} = \frac{\partial q_k(t)}{\partial \tau_j(t)} \frac{\partial \tau_j(t)}{\partial \tau_{Fj}^*(t)} \frac{\partial \tau_{Fj}^*(t)}{\partial w_{ij}(t)} \tag{18}$$

Since $\tau_j(t) = \tau_{Fj}^*(t) + \tau_{rj}(t)$ and from the fact of (12), we have

$$\frac{\partial q_k(t)}{\partial w_{ij}(t)} = \frac{\partial q_k(t)}{\partial \tau_j(t)} \frac{h_i}{\sum_{i=1}^r h_i} \tag{19}$$

Therefore,

$$w_{ij}(t+1) = w_{ij}(t) + \eta \sum_{k=1}^{3} e_k(t) \frac{\partial q_k(t)}{\partial \tau_j(t)} \frac{h_i}{\sum_{i=1}^r h_i} \tag{20}$$

C. An Approximate Evaluation of Output Jacobian with Respect to Input Torque

For the simplicity of the problem, let us consider the case of $d \simeq 0$. Then, it is easily found, from the dynamical model of the slave robot [2] or [7], that

$$M^{-1} = \begin{bmatrix} \frac{1}{m} & 0 & 0 \\ 0 & \frac{1}{m} & 0 \\ 0 & 0 & \frac{1}{I} \end{bmatrix}, \; M^{-1}B(\boldsymbol{q}) = \begin{bmatrix} \frac{1}{mr}\cos\theta & \frac{1}{mr}\cos\theta \\ \frac{1}{mr}\sin\theta & \frac{1}{mr}\sin\theta \\ \frac{1}{Ir}R & -\frac{1}{Ir}R \end{bmatrix} \tag{21}$$

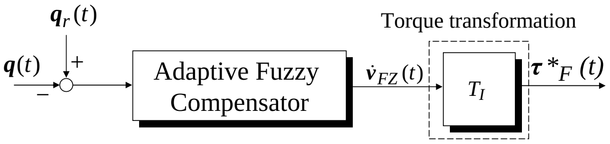

Fig. 7. An adaptive fuzzy compensator with a torque transformation

Therefore, the evaluation of $\partial\ddot{\boldsymbol{q}}/\partial\tau_i(t)$ is equivalent to

$$\frac{\partial M^{-1}B(\boldsymbol{q})\boldsymbol{\tau}(t)}{\partial\tau_1(t)} = \begin{bmatrix} \dfrac{\cos\theta}{mr} \\ \dfrac{\sin\theta}{mr} \\ \dfrac{R}{Ir} \end{bmatrix}, \quad \frac{\partial M^{-1}B(\boldsymbol{q})\boldsymbol{\tau}(t)}{\partial\tau_2(t)} = \begin{bmatrix} \dfrac{\cos\theta}{mr} \\ \dfrac{\sin\theta}{mr} \\ -\dfrac{R}{Ir} \end{bmatrix} \tag{22}$$

Now, using an approximation of $\dot{\boldsymbol{q}} = [\boldsymbol{q}(t) - \boldsymbol{q}(t-1)]/\Delta t$ and noting that the evaluation of $\partial\boldsymbol{q}(t)/\partial\tau_j(t)$ is equivalent to $\partial(\Delta t)^2 M^{-1}B(\boldsymbol{q})\boldsymbol{\tau}(t)/\partial\tau_j(t)$, we have

$$\frac{\partial\boldsymbol{q}(t)}{\partial\tau_1(t)} = \begin{bmatrix} (\Delta t)^2\cos\theta \\ (\Delta t)^2\sin\theta \\ (\Delta t)^2 R_{mI}R \end{bmatrix}, \quad \frac{\partial\boldsymbol{q}(t)}{\partial\tau_2(t)} = \begin{bmatrix} (\Delta t)^2\cos\theta \\ (\Delta t)^2\sin\theta \\ -(\Delta t)^2 R_{mI}R \end{bmatrix} \tag{23}$$

where a new learning rate should be interpreted as $\eta/mr \triangleq \eta'$, and the ratio of mass to the moment of inertia $R_{mI} \triangleq m/I$ and the tread $2R$ are assumed to be known to use the above evaluation of the output Jacobian with respect to the input torque in the online learning.

D. A Transformation of Fuzzy Compensator

If we have an acceleration reasoning from a fuzzy compensator as shown in Fig. 7, then we need a torque transformation matrix $T_I \triangleq \{T_{Iij}\},\ i,j = 1,2$ to obtain $\boldsymbol{\tau}^*_F(t) = T_I\dot{\boldsymbol{v}}_{FZ}(t)$.

Therefore we obtain

$$\frac{\partial q_k(t)}{\partial w_{ij}(t)} = \frac{\partial q_k(t)}{\partial\tau_j(t)}\frac{\partial\tau_j(t)}{\partial\tau^*_{Fj}(t)}\frac{\partial\tau^*_{Fj}(t)}{\partial\dot{v}_{FZj}(t)}\frac{\partial\dot{v}_{FZj}(t)}{\partial w_{ij}(t)} \tag{24}$$

$$\frac{\partial\tau^*_{Fj}(t)}{\partial\dot{v}_{FZj}(t)} = T_{Ijj} \tag{25}$$

so that it follows that

$$\frac{\partial q_k(t)}{\partial w_{ij}(t)} = \frac{\partial q_k(t)}{\partial\tau_j(t)}T_{Ijj}\frac{h_i}{\sum_{i=1}^r h_i} \tag{26}$$

In the sequel,

$$w_{ij}(t+1) = w_{ij}(t) + \eta\sum_{k=1}^{3} e_k(t)\frac{\partial q_k(t)}{\partial\tau_j(t)}T_{Ijj}\frac{h_i}{\sum_{i=1}^r h_i} \tag{27}$$

IV. Learning of Antecedent Part

Similarly, we can derive the update algorithms for the design parameters in the antecedent part, i.e., $c_{ij}(t)$ and $w_{dij}(t)$.

A. Case of $c_{ij}(t)$

$$c_{ij}(t+1) = c_{ij}(t) - \eta\frac{\partial J}{\partial c_{ij}(t)} \tag{28}$$

Here, it follows that

$$\frac{\partial J}{\partial c_{ij}(t)} = \sum_{k=1}^{3}\frac{\partial J}{\partial e_k(t)}\frac{\partial e_k(t)}{\partial c_{ij}(t)} \tag{29}$$

$$= \sum_{k=1}^{3} e_k(t)\frac{\partial e_k(t)}{\partial q_k(t)}\frac{\partial q_k(t)}{\partial c_{ij}(t)} \tag{30}$$

$$= -\sum_{k=1}^{3} e_k(t)\frac{\partial q_k(t)}{\partial c_{ij}(t)} \tag{31}$$

$$\frac{\partial q_k(t)}{\partial c_{ij}(t)} = \frac{\partial q_k(t)}{\partial\tau_j(t)}\frac{\partial\tau_j(t)}{\partial\tau^*_{Fj}(t)}\frac{\partial\tau^*_{Fj}(t)}{\partial h_i(t)}\frac{\partial h_i(t)}{\partial\mu_{A_{ij}}(e_j)} \times\frac{\partial\mu_{A_{ij}}(e_j)}{\partial c_{ij}(t)} \tag{32}$$

Since

$$\frac{\partial\tau_j(t)}{\partial\tau^*_{Fj}(t)} = 1 \tag{33}$$

$$\frac{\partial\tau^*_{Fj}(t)}{\partial h_i(t)} = \frac{w_{ij}}{\sum_{i=1}^r h_i} - \frac{\sum_{i=1}^r h_i w_{ij}}{\left(\sum_{i=1}^r h_i\right)^2} \tag{34}$$

$$\frac{\partial h_i(t)}{\partial\mu_{A_{ij}}(e_j)} = \prod_{j=1,i\neq j}^{n}\mu_{ij}(e_j) \tag{35}$$

$$\frac{\partial\mu_{A_{ij}}(e_j)}{\partial c_{ij}(t)} = (-2\ln(0.5)(e_j - c_{ij})w_{dij}^2) \times\exp\{\ln(0.5)(e_j - c_{ij})^2 w_{dij}^2\} \tag{36}$$

we have

$$\frac{\partial q_k(t)}{\partial c_{ij}(t)} = \frac{\partial q_k(t)}{\partial\tau_j(t)}f_c(w_{ij}, e_j, c_{ij}, w_{dij}) \tag{37}$$

in which

$$f_c(w_{ij}, e_j, c_{ij}, w_{dij}) = \big[(-2\ln(0.5)(e_j - c_{ij})w_{dij}^2) \times\exp\{\ln(0.5)(e_j - c_{ij})^2 w_{dij}^2\}\big] \times\left[\frac{w_{ij}}{\sum_{i=1}^r h_i} - \frac{\sum_{i=1}^r h_i w_{ij}}{\left(\sum_{i=1}^r h_i\right)^2}\right] \times\prod_{j=1,i\neq j}^{n}\mu_{ij}(e_j) \tag{38}$$

Therefore,

$$c_{ij}(t+1) = c_{ij}(t) + \eta\sum_{k=1}^{3} e_k(t)\frac{\partial q_k(t)}{\partial\tau_j(t)}f_c(w_{ij}, e_j, c_{ij}, w_{dij}) \tag{39}$$

B. Case of $w_{dij}(t)$

$$w_{dij}(t+1) = w_{dij}(t) - \eta \frac{\partial J}{\partial w_{dij}(t)} \quad (40)$$

Here, it follows that

$$\begin{aligned} \frac{\partial J}{\partial w_{dij}(t)} &= \sum_{k=1}^{3} \frac{\partial J}{\partial e_k(t)} \frac{\partial e_k(t)}{\partial w_{dij}(t)} & (41) \\ &= \sum_{k=1}^{3} e_k(t) \frac{\partial e_k(t)}{\partial q_k(t)} \frac{\partial q_k(t)}{\partial w_{dij}(t)} & (42) \\ &= -\sum_{k=1}^{3} e_k(t) \frac{\partial q_k(t)}{\partial w_{dij}(t)} & (43) \end{aligned}$$

$$\begin{aligned} \frac{\partial q_k(t)}{\partial w_{dij}(t)} &= \frac{\partial q_k(t)}{\partial \tau_j(t)} \frac{\partial \tau_j(t)}{\partial \tau^*_{Fj}(t)} \frac{\partial \tau^*_{Fj}(t)}{\partial h_i(t)} \frac{\partial h_i(t)}{\partial \mu_{A_{ij}}(e_j)} \\ &\quad \times \frac{\partial \mu_{A_{ij}}(e_j)}{\partial w_{dij}(t)} & (44) \end{aligned}$$

Since

$$\begin{aligned} \frac{\partial \mu_{A_{ij}}(e_j)}{\partial w_{dij}(t)} &= (2\ln(0.5)(e_j - c_{ij})^2 w_{dij}) \\ &\quad \times \exp\{\ln(0.5)(e_j - c_{ij})^2 w_{dij}^2\} & (45) \end{aligned}$$

we have

$$\frac{\partial q_k(t)}{\partial w_{dij}(t)} = \frac{\partial q_k(t)}{\partial \tau_j(t)} f_{w_d}(w_{ij}, e_j, c_{ij}, w_{dij}) \quad (46)$$

in which

$$\begin{aligned} f_{w_d}(w_{ij}, e_j, c_{ij}, w_{dij}) &= \big[(2\ln(0.5)(e_j - c_{ij})^2 w_{dij}) \\ &\quad \times \exp\{\ln(0.5)(e_j - c_{ij})^2 w_{dij}^2\}\big] \\ &\quad \times \left[\frac{w_{ij}}{\sum_{i=1}^{r} h_i} - \frac{\sum_{i=1}^{r} h_i w_{ij}}{\left(\sum_{i=1}^{r} h_i\right)^2}\right] \\ &\quad \times \prod_{j=1, i\neq j}^{n} \mu_{ij}(e_j) & (47) \end{aligned}$$

Therefore,

$$\begin{aligned} w_{dij}(t+1) &= w_{dij}(t) \\ &\quad + \eta \sum_{k=1}^{3} e_k(t) \frac{\partial q_k(t)}{\partial \tau_j(t)} f_{w_d}(w_{ij}, e_j, c_{ij}, w_{dij}) \\ & & (48) \end{aligned}$$

V. SIMULATIONS

The dynamical model given in Izumi et al. [2] or Fierro and Lewis [7] as a slave robot, is now transformed into a steering model for controlling a master robot. That is, an actual model of Eq. (2) as a master robot can now be reduced to the following equation:

$$\begin{bmatrix} m & 0 \\ 0 & I - md^2 \end{bmatrix} \begin{bmatrix} \dot{v} \\ \ddot{\theta} \end{bmatrix} + \boldsymbol{\tau}_d = \frac{1}{r} \begin{bmatrix} 1 & 1 \\ R & -R \end{bmatrix} \begin{bmatrix} \tau_r \\ \tau_l \end{bmatrix} \quad (49)$$

where m denotes the mass of the robot, d is the offset distance of steering axis, I is the moment of inertia of the robot, $2R$ denotes the tread of the robot, and r denotes the radius of wheel.

We conducted the training of NN to obtain an inverse mapping of the virtual master robot and after that implemented the neurointerface together with a feedback compensator for controlling the actual (slave) robot. The steering model of Eq. (2) and the slave model given in Izumi et al. [2] were all simulated by using a simple Euler's method, under the condition of the sampling width $\Delta t = 0.02$ [s].

When training the NN, known friction inputs as disturbance torques were supplied to the NN, where $\{\tau_{1d}, \tau_{2d}\} = \{0.005, 0.005\}$[Nm] for all simulations. Note also that the initial values of connection weights w_i for the NN were set by using uniform random numbers.

We collected the input-output data from the steering model with deterministic input torques, as the training data for the NN. That is, it was assumed that the torque inputs were generated by using sinusoidal functions [6].

In the following, we considered a case where the dynamical and kinematic parameters except for the offset distance of steering axis d were all unknown.

A. A PD Feedback Compensation

We first tested for untrained reference by applying the neurointerface with a PD feedback compensator, where the PD feedback gains were set to $k_{ip} = 1$ and $k_{id} = 2$ for $i = 1, 2, 3,$ through trial and error. The resultant test trajectories are shown in Fig. 8.

It is found from this figure that as expected, a simple use of only the NN-based feedforward controller has some output deviations in Cartesian coordinate due to the mapping errors of NN, whereas adding a PD feedback compensator gives a better response.

B. An Adaptive Fuzzy Compensation

The second tested was implemented with the neurointerface together with a PD-based feedback controller used above and a fuzzy feedback compensator, where the constant parameters in the consequent part of reasoning were learned online with a learning rate η of 0.1, i.e., the center value c_{ij} and the reciprocal value of the deviation were all fixed in advance. The resultant test trajectories are shown in Fig. 9.

It is found from this figure that as expected, a simple use of only the NN-based feedforward controller and a PD feedback compensator has some output deviations in Cartesian coordinate, whereas adding an adaptive fuzzy feedback compensator gives a better response.

VI. CONCLUSIONS

We have presented a design method for a fuzzy compensator in constructing an NN-based feedforward controller, i.e., neurointerface, by applying a concept of a virtual master-slave system. Since there was a practical mapping error of NN, a PD-based feedback compensator or fuzzy compensator was further added to the neurointerface to suppress the output deviations. The effectiveness of the proposed method was

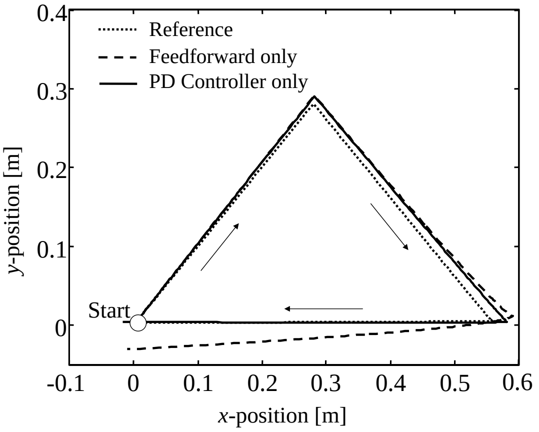

Fig. 8. A tested reference using a neurointerface with a PD feedback co

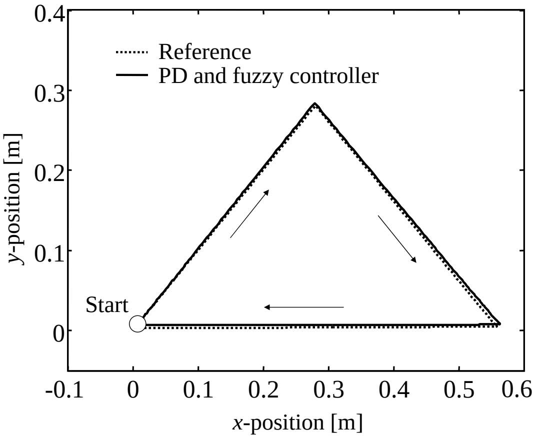

Fig. 9. A tested reference using a neurointerface with fuzzy compensator and PD feedback controller

shown through a simple simulation for solving a trajectory tracking control problem of a nonholonomic mobile robot with two-independent driving wheels.

We have to further investigate a more general adaptive fuzzy compensator, learning all of design parameters in the fuzzy reasoning, to obtain satisfactory results against any variational deviations caused by the changes of the robot's mass or against any external disturbances.

REFERENCES

[1] W. Widrow and M. L. Lamego, Neurointerfaces, *IEEE Trans. on Control System Technology*, vol. 10, no. 2, pp. 221–228, March 2002.

[2] K. Izumi, R. Syam, and K. Watanabe, "Neural network based disturbance canceller with feedback error learning for nonholonomic mobile robots," in *Proc. of the 4th Int. Symposium on AI System (ISIS 2003)*, Sept. 25–28, 2003, Jeju, Korea, pp. 443–446.

[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] R. Syam, K. Watanabe, and K. Izumi, "A study on constructing a neuro-interface using the concept of virtual master-slave system," in *Proc. of the 9th Int. Symposium on Artificial Life and Robotics (AROB 9th '04)*, Jan. 28–30, 2004, Beppu, Oita, Japan, vol.2, pp. 527–530.

[5] R. Syam, K. Watanabe, and K. Izumi, "A neurointerface using a concept of virtual master-slave for controlling nonholonomic mobile robots," in *Proc. of Int. Workshop on Fuzzy Systems & Innovational Computing 2004 (FIC2004)*, June 2–3, 2004, Kitakyushu, Japan.

[6] R. Syam, K. Watanabe, and K. Izumi, "Concept of Virtual Master-Slave Systems and Its Application to the Design of a Neuro-Interface," in *Proc. of the 43rd SICE Annual Conference (SICE2004)*, August 4–6, 2004, Sapporo, Japan.

[7] R. Fierro and F. L. Lewis, "Control of nonholonomic mobile robot using neural networks," *IEEE Trans. on Neural Networks*, vol. 9, no. 4, pp. 589–600, 1998.

[8] M. Teshnehlab and K. Watanabe, *Intelligent Control Based on Flexible Neural Networks*, Dordrecht, The Netherlands: Kluwer Academic Publishers, 1999, pp. 47–49.

[9] K. Watanabe, T. Fukuda, and S. G. Tzafestas, "An adaptive control for CARMA systems using linear neural networks," *Int. J. Control*, vol. 56, no. 2, pp. 483–497, 1992.