PATH TRACKING CONTROL OF A MOBILE ROBOT BY A LINEARIZATION USING RBFS NEURAL NETWORKS

Hidetomo Ichihashi , Tetsuya Miyoshi and Kazunori Nagasaka

Department of Industrial Engineering
University of Osaka Prefecture
Sakai, Osaka 593, JAPAN

ABSTRACT

In this paper, we propose a method to linearize nonlinear systems, which stabilizes the nonlinear systems so long as the learning successfully converges. To obtain a coordinate transformation from nonlinear systems to linear systems, Gaussian Radial Basis Functions networks(RBFs) which can be reinterpreted as three layered neural networks and fuzzy models, are used. In numerical simulations we show the effectiveness of the proposed linearization method.

INTRODUCTION

In this paper we propose a linearization method of nonlinear systems and apply a linear control scheme for stabilizing the nonlinear system. To obtain a coordinate transformation from nonlinear systems to linear systems, Gaussian Radial Basis Functions networks(RBFs) (Moody *et al.*,1989) are used.

A path tracking control of mobile robots, using the exact linearization and the time scale transformation, was developed by Sampei *et al.*(1989). Nakamura *et al.*(1993) showed, when the sequence of proper straight lines and circle arcs are given as the desired path, the trailer vehicle can track the desired path. This method is based on the locally linear approximation of the system. Both of the above methods are the model based control and therefore need the mathematical model of the controlled object.

A neural controller of truck and trailer was developed by Nguyen *et al.*(1990). Though the model free optimal feedback control is realized by their approach, thousands of backups are required for training the neural network. In this paper, we propose a model free method to linearize nonlinear systems, which stabilizes nonlinear systems so long as the learning successfully converges.

LINEARIZATION BY RBFS NETWORKS

Let δ be a manipulated variable and $\boldsymbol{\theta} = (\theta_1, \theta_2, \cdots, \theta_n)^T$ be a vector of state variables , where T denotes transpose. Let us assume that a nonlinear system is described as:

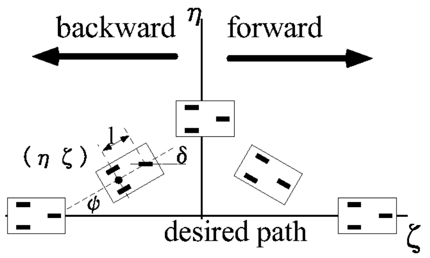
$$\dot{\boldsymbol{\theta}} \;=\; \boldsymbol{f}_1(\boldsymbol{\theta}) + \boldsymbol{f}_2(\boldsymbol{\theta}) \cdot f_3(\delta) \qquad (1)$$

where $\boldsymbol{f}_1$, $\boldsymbol{f}_2$ are differentiable functions.

By the exact linearizing method and time scale transformation (Su, 1982)(Sampei *et al.*, 1989) this nonlinear system is linearized into the following :

$$\frac{d\boldsymbol{x}}{dt} = \begin{pmatrix} 0 & 1 & 0 & \cdots & 0 \\ 0 & 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \ddots & \ddots & \vdots \\ \vdots & \vdots & \ddots & \ddots & 1 \\ 0 & 0 & 0 & \cdots & 0 \end{pmatrix} \boldsymbol{x} + \begin{pmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ 1 \end{pmatrix} u \qquad (2)$$

In this paper we deal with a three wheeled mobile robot as a nonlinear system and assume that the states of the system are directly observable. We will assume that the robot moves slowly and we can ignore the slide slips. In Fig. 1, a diagram of the mobile robot is shown. The object of this paper is to propose a method to linearize the nonlinear system using RBFs networks and to make the mobile robot follow a desired path(ζ axis



ψ : angle of the robot with horizontal
δ : steering angle
l : length of the robot
(η, ζ) : Cartesian coordinate of the robot

Figure 1: A three wheeled mobile robot

in Fig. 1) as it moves forward or backward. The linear controller makes the state vector $\boldsymbol{\theta}$ of the mobile robot go to zero because the nonlinear system is transformed into a linear system which can be stabilized by the linear controller.

Let the state equation of the three wheeled mobile robot be unknown , and the vector of state variables be $\boldsymbol{\theta} = (\psi, \eta)^T$. Using the approximate linearization proposed in this paper, this nonlinear system can be transformed into a linear system as :

$$\dot{\boldsymbol{x}} = A\boldsymbol{x} + Bu \tag{3}$$

where

$$A = \begin{pmatrix} 0 & 1 \\ 0 & 0 \end{pmatrix}, \qquad B = (0,1)^T, \boldsymbol{x} \ = (x_1, x_2)^T$$

This linear system (A, B) is controlable.

Let the transformation from the nonlinear system into the linear system be represented by Gaussian RBFs networks $g_1(\boldsymbol{\theta})$ and $g_2(\boldsymbol{\theta})$. Then,

$$\Theta = \left\{\boldsymbol{\theta} = (\psi, \eta) \middle| \psi \in (-\pi, \pi), \eta \in (-5l, 5l)\right\} \tag{4}$$

is transformed into

$$X = \{\boldsymbol{x} = (x_1, x_2) | x_1, x_2 \in (-1, 1)\} \tag{5}$$

by g_1 and g_2 as:

$$\boldsymbol{x} = \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \begin{pmatrix} g_1(\psi, \eta) \\ g_2(\psi, \eta) \end{pmatrix} \tag{6}$$

g_1 and g_2 are the coordinate transformation such that $\boldsymbol{x}$ tends to zero as $\boldsymbol{\theta}$ tends to zero.

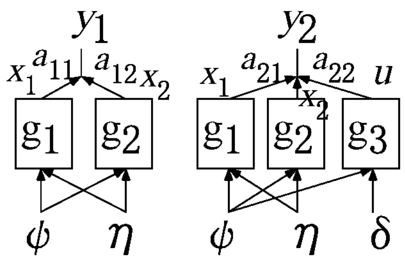


Figure 2: Coordinate transformation network which consists of RBFs

The transformation between the manipulated variable (steering angle δ) of the controlled object and the manipulated variable (u) of the linearized system is represented by Gaussian RBFs g_3 and g_4 as:

$$u = g_3(\psi, \delta) \tag{7}$$

$$\delta = g_4(\psi, u) \tag{8}$$

The functional relation between u and δ is restricted to be strictly increasing.

g_1, g_2, g_3 and g_4 are defined by two dimensional Gaussian RBFs neural networks as:

$$g_i(z_1, z_2) = \sum_{k=1}^{K} \mu_k^i(z_1, z_2) w_k^i \qquad (i = 1, 2, 3, 4) \tag{9}$$

$$\mu_k^i(z_1, z_2) = \prod_{j=1}^{2} \exp\left(-\frac{(z_j - a_{kj}^i)^2}{b_{kj}^i}\right) \tag{10}$$

The RBFs networks can be regarded as both three layered neural networks (Moody *et al.*, 1989) (Poggio *et al.*, 1990) and simplified fuzzy reasoning models (Ichihashi *et al.*, 1993).

In Fig. 2, the networks for the transformation is shown. In Fig. 2, $a_{11} = 0$, $a_{12} = 1$, $a_{21} = 0$, $a_{22} = 0$ are the elements of matrix A of Eq. (3). The outputs of the networks are

$$\boldsymbol{y} = (y_1, y_2) = (g_2(\psi, \eta), g_3(\psi, \delta)) \tag{11}$$

where y_1 and y_2 represent $\dot{x_1}$ and $\dot{x_2}$ of Eq. (3) respectively .

The training of networks is carried out for a wide variety of positional states (ψ, η) and steering angles(δ). The RBFs networks are trained so as to minimize the following cost function as:

$$E_1 = \sum \|\dot{\boldsymbol{g}} - \boldsymbol{y}\|^2$$

$$+k_1 \sum H(\varepsilon - \frac{\partial u}{\partial \delta})(\varepsilon - \frac{\partial u}{\partial \delta})^2$$

$$+k_2 \|\boldsymbol{x}\|^2_{\boldsymbol{\theta}=0} \tag{12}$$

$$E_2 = \sum \{\delta - g_4(\psi, u)\}^2 \tag{13}$$

where ε is a positive small number and

$$H(z) = \begin{cases} 0 & ; z \leq 0 \\ 1 & ; z > 0 \end{cases} \tag{14}$$

In Eq.(12) the constants k_1 and k_2 are the penalty coefficients for respective error terms. The summations in Eqs.(12) and (13) are taken for all selected values of positional states and steering angles. $\dot{\boldsymbol{g}}$ in Eq.(12) is calculated as

$$\dot{g_m} = \frac{\partial g_m}{\partial \psi}\frac{d\psi}{dt} + \frac{\partial g_m}{\partial \eta}\frac{d\eta}{dt} \quad (m = 1, 2) \tag{15}$$

where $d\psi/dt$ and $d\eta/dt$ are observable from the mobile robot before learning. The learning rule is based on the gradient descent method such as :

$$w_k^{mNEW} = w_k^{mOLD} - \tau \frac{\partial E_1}{\partial w_k^m} \tag{16}$$

$$a_{ki}^{mNEW} = a_{ki}^{mOLD} - \tau \frac{\partial E_1}{\partial a_{ki}^m} \tag{17}$$

$$b_{ki}^{mNEW} = b_{ki}^{mOLD} - \tau \frac{\partial E_1}{\partial b_{ki}^m} \tag{18}$$

where τ is a positive learning rate.

CONTROLLER DESIGN

The state vector and the input of the nonlinear system are transformed into those of the linear system by the RBFs networks (g_1, g_2, g_3) after learning. We design a linear controller by defining poles for the linearized system, so as to make the linear system stable. Let the state feedback for the linearized system be

$$u = \boldsymbol{cx} = c_1x_1 + c_2x_2 \tag{19}$$

where $\boldsymbol{c}$ is the feedback gain. By the inverse transformation $\delta = g_4(\psi, u)$, $\boldsymbol{\theta}$ tends to zero as $\boldsymbol{x}$ tend to zero, that is, δ makes the mobile robot follow the desired path.

We can design a servo controller for the linearized

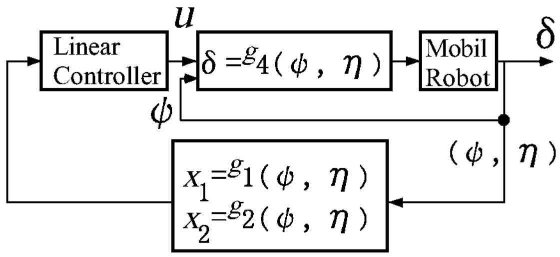


Figure 3: A state feedback controller

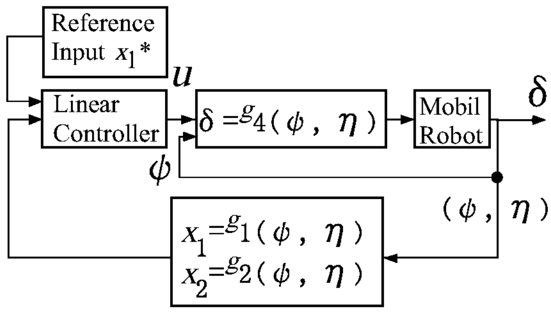


Figure 4: A servo controller

system . Let the feedback for the linear system be

$$u = c_1x_1 + c_2x_2 + c_3 \int_0^t (x_1 - x_1^*)dt \tag{20}$$

where reference input x_1^* is the state value of the linearized system without error. x_1^* can be calculated when its initial value is obtained from the initial values of ψ and η by g_1 and g_2. The servo controller consists of g_1 , g_2 and g_4 as shown in Fig. 4. This controller eliminates steady-state position error, when there exists constant error in the steering angle.

SIMULATION

The values of τ, k_1, k_2, l, v and K are set to $\tau = 0.000003$, $k_1 = k_2 = 1.0$, $l = 1.0(m)$, $v = 1.0(m/s)$ and $K = 9$. The initial values of w_k^m and b_k^m are $w_k^m = -1.0$ $(k = 1)$, $w_k^m = -0.5$ $(k = 2, 4)$, $w_k^m = 0.0$ $(k = 3, 5, 7)$, $w_k^m = 0.5$ $(k = 6, 8)$, $w_k^m = 1.0$ $(k = 9)$ and $b_{ki}^m = 0.5$ $(k = 1, \cdots, 9)$. The initial value of the cost function Eq.(12) was 4030 and the value after 5000 training iterations was 26.4 .

Fig. 5 (a) and (b) show the simulation results where the robot moves forward and backward respectively. Note that the robot , from any initial position within

(a) Forward movement

(b) Backward movement

Figure 5: Forward and backward movements of the mobile robot

(a) By state feedback controller

(b) By servo controller

Figure 6: Forward movement with constant error(-0.03 rad) in steering angle δ

$-\pi \leq \psi \leq \pi$, can follow the desired path . In the learning of RBFs networks, the value of ψ is chosen from $[-\pi, \pi]$ and only the forward movement is taken into account.

Fig. 6 shows the result in the case that the steering angle δ has constant error(-0.03 rad). As shown in Fig.6 (a) , the state feedback u in Eq.(19) cannot make the mobile robot follow the desired path exactly. However, Fig.6 (b) shows that the servo controller can eliminate state position error and makes the robot follow the desired path without error.

CONCLUSION

We have developed an approximate linearization method using Gaussian RBFs neural networks. By the stabilizing control of the linearized system, the mobile robot tracks the desired path even when it moves backward. Furthermore the robot with constant error in the steering angle tracks the desired path without error by a servo controller. An application to a trailer truck backing up control is currently carried on.

References

[1] Ichihashi,H. and Tokunaga,M. , 1993, "Neuro-Fuzzy Optimal Control of Backing up a Trailer Truck", *Proceedings, IEEE-ICNN'93 (San Francisco)*, Vol.1, pp.306-311.

[2] Moody,J. and Darken,C.J., 1989, "Fast Learning in Networks of Locally-Tuned Processing Unit", *Neural Computation*, Vol.1, pp.281-294.

[3] Nakamura,M. and Yuta,S., 1993, "Trajectory Control of Trailer Type Mobile Robots", *Proceedings, IEEE/RSJ International Workshop on Intelligent Robot and Systems*, pp.2257-2263.

[4] Nguyen,D.H. and Widrow,B.,1990, " Neural Networks for Self-Learning Control System", *IEEE Control Systems Magazine* , April, pp.18-23.

[5] Poggio,T. and Girosi,F., 1990, "Regularization Algorithms for Learning That Are Equivalent to Multilayer Networks", *Sciences*, Vol.247, pp.987-982.

[6] Sampei,M., Tamura,T., Itoh,T. and Nakamichi,M., 1989, "Path Tracking Control of Trailer-Like Mobile Robot", *Proceedings, IEEE/RSJ International Workshop on Intelligent Robot and Systems*, pp.193-198.

[7] Su,R., 1982, "On the Linear Equivalents of Nonlinear Systems", *Systems and Control Letters*, Vol.2, No.1, pp.48-52.