'94 KACC (1994. 10. 17 ~ 20)

A SELF-LEARNING FUZZY CONTROLLER WITH A FUZZY APPROXIMATION OF INVERSE MODELING

Y. R. Seo°* and C. H. Chung*
Dep. of Control & Instrumentation Eng., Kwangwoon University*

Abstract In this paper, a self-learning fuzzy controller is designed with a fuzzy approximation of an inverse model. The aim of an identification is to find an input command which is control of a system output. It is intuitional and easy to use a classical adaptive inverse modeling method for the identification, but it is difficult and complex to implement it. This problem can be solved with a fuzzy approximation of an inverse modeling. The fuzzy logic effectively represents the complex phenomena of the real world. Also fuzzy system could be represented by the neural network that is useful for a learning structure. The rule of a fuzzy inverse model is modified by the gradient descent method.

The goal is to be obtained that makes the design of fuzzy controller less complex, and then this self-learning fuzzy controller can be used for nonlinear dynamic system. We have applied this scheme to a nonlinear Ball and Beam system.

1. INTRODUCTION

A fuzzy control is composed as fuzzification, inference engine, rule base, and defuzzification. The fuzzy controller has control of a plant with a fuzzy inference logic. The inference rules expressing the input-output relation of data are modified with a learning algorithm.

The gradient descent learning algorithm makes it possible to train the neural network identifiers (Narendra *et al.*, 1990) on-line to match unknown nonlinear mappings. Fuzzy controller can be represented as feedforward network model. On the basis of this point, a gradient descent algorithm was developed in (Wang *et al.*, 1992), to train the fuzzy controller to match input-output data pairs. To use fuzzy system as approximation of an inverse model for nonlinear dynamic system, we need to know how to choose their parameters such that they perform the desired nonlinear mapping. With respect to its design parameters, the idea in the gradient descent learning algorithm is to use the chain rule to determine the gradient of the output errors of the fuzzy system.

For on-line learning fuzzy controller about unknown nonlinear system, we use an inverse modeling technique. That is, based on approach for the classical adaptive control of liner and nonlinear systems with unknown dynamics, proposed (Widrow *et al.*, 1985). Unfortunately adaptive inverse modeling technique is difficult and complex to implement.

The fuzzy approximation of the inverse modeling is easy to design and has a good performance. The fuzzy system is capable of approximating any real continuous function on a compact set to arbitrary accuracy (Wang, 1992).

2. FUZZY CONTROLLER

2.1 General Element of Fuzzy Controller

In this paper, we consider a fuzzy control system (Lee, 1990a,b) whose general configuration is shown in Fig. 1. There are four basic elements in such a fuzzy system: fuzzifier, fuzzy rule base, fuzzy rule inference engine, and defuzzifier. We consider multi-input, single-output fuzzy system: $U \subset R^n \rightarrow R$, where U is compact.

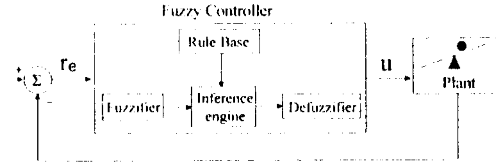

Fig. 1. General configuration of fuzzy control system.

2.1.1 Fuzzifier

The fuzzifier performs a mapping from the observed crisp input space $U \subset R^n$ to the fuzzy set defined in U, where a fuzzy set(Zadeh, 1965) defined in U is characterized by a membership function $\mu_F : U \rightarrow [0,1]$, and is labeled by a linguistic term F.

2.1.2 Fuzzy Rule Base

The fuzzy rule base consists of a set of linguistic rule in the form of "IF a set of conditions is satisfied, THEN a set of consequences is inferred." We consider the case where the fuzzy rule base consists of m rules and n variables in the following form:

$$\textbf{Rule}_j: \text{ IF } x_1 \text{ is } A_1^j \text{ and } x_2 \text{ is } A_2^j \text{ and} \cdots \text{and } x_n \text{ is } A_n^j, \text{ THEN } z \text{ is } B^j$$

2.1.3 Fuzzy Inference Engine

The fuzzy inference engine is decision-making logic that employs fuzzy rules from the fuzzy rule base to determine a mapping from the fuzzy sets in the input space U to the fuzzy set in the output space R.

2.1.4 Defuzzifier

The defuzzifier performs a mapping from the fuzzy sets in R to crisp point in R.

2.2 Fuzzy Controller

For simplicity, we assume the fuzzy controller under consideration has two input variables and one output. Each input variable has two fuzzy sets defined by membership function in its universe of discourse and four fuzzy rules are made from. If this fuzzy system uses product-inference logic and singleton defuzzifier for output variable, it can be described by a network model in Fig.2 (Seo, E.T. 1994).

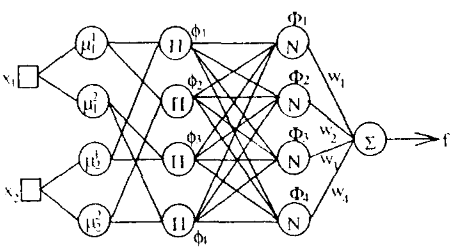

Fig 2. Fuzzy controller as a feedforward network model.

The output of this fuzzy controller is

$$f(x) = \sum_{k=1}^{4} \Phi_k \cdot w_k \qquad (1)$$

where

$$\Phi_k = \frac{\phi_k}{\sum_{i=1}^{4} \phi_i}, \qquad (2)$$

and

$$\begin{aligned} \phi_1(x) &= \mu_1^1(x_1) \cdot \mu_2^1(x_2), \\ \phi_2(x) &= \mu_1^1(x_1) \cdot \mu_2^2(x_2), \\ \phi_3(x) &= \mu_1^2(x_1) \cdot \mu_2^1(x_2), \\ \phi_4(x) &= \mu_1^2(x_1) \cdot \mu_2^2(x_2). \end{aligned} \qquad (3)$$

If we choose the parameters of (1) for a certain purpose, this fuzzy controller behaves properly as a filter, i.e., well-defined fuzzy controller is realizable. If all parameters determining the membership functions of input variables are fixed in (1), the only remaining design parameters are w_k and the fuzzy controller is linear for these parameters. By adapting this point of view, we are able to use some very efficient linear parameter estimation methods, e.g., gradient descent method, and least square method, etc.

3. ADAPTIVE INVERSE MODELING

3.1 Adaptive Inverse Modeling

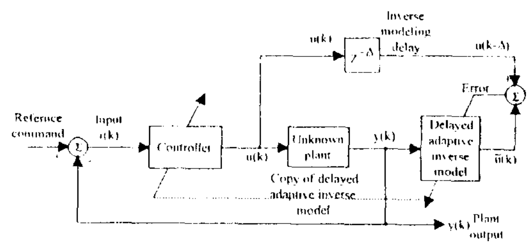

Fig. 3. Adaptive inverse model control system.

An unknown plant can be made to track an input command signal when this signal is applied to a controller whose transfer function approximates the inverse of its transfer function. The output becomes a driving signal of the plant. In the adaptive inverse control method, the parameters of the controller are obtained by an adaptive inverse modeling process applied to the plant. Consequently, after the adaptation phase, the plant output will track the reference command signal.

$$y(k)_{\text{plant output}} \cong r(k-\Delta)_{\text{input command}} \qquad (4)$$

Since the adaptive inverse control system is obtained from the cascade of the adaptation phase, Fig. 3 shows the plant and the inverse model. This method is produced an inverse model of an unknown plant. Consequently, that is effected a equalization or a deconvolution by inverse model.

4. GRADIENT DESCENT LEARNING for a FUZZY APPROXIMATION of INVERSE MODELING

4.1 Gradient Descent Learning

In Fig. 3, we have defined an approximation error;

$$e_k = [u(k-\Delta) - \tilde{u}(k)] \tag{5}$$

In gradient descent algorithm(Widrow *et al.*, 1985), we would estimate the gradient of $MSE = E[e_k^2] \approx e_k^2$. At each iteration in the adaptive process, we have a gradient estimate of the form

$$\hat{\nabla}_k = \frac{\partial \varepsilon_k^2}{\partial w} = 2\varepsilon_k \frac{\partial \varepsilon_k}{\partial w} \tag{6}$$

where w is the weight of $\tilde{u}(k)$ and k is the state of a adaptive process system. With this simple estimate of the gradient, we can specify a steepest descent type algorithm as follows

$$w_{k+1} = w_k - \rho \hat{\nabla}_k \tag{7}$$

ρ is the gain constant that regulates the speed and stability of adaptation.

4.2 Fuzzy Approximation of Inverse Modeling

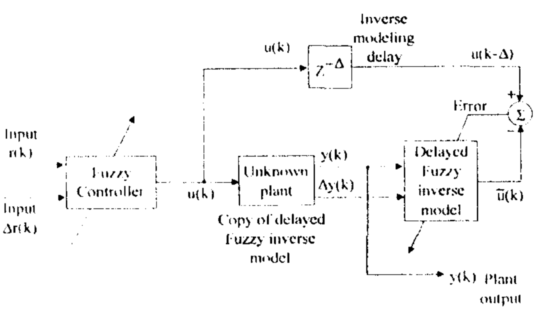

Fig. 4. Self-learning fuzzy controller with fuzzy inverse model.

The fuzzy controller with the fuzzy approximation of inverse model is obtained by incorporating the gradient descent learning system. The inputs are the reference signal $r(k)$ and its change $\Delta r(k)$ for the fuzzy controller, and plant output $y(k)$ and its variation $\Delta y(k)$ for the inverse model(Fig. 4), where

$$\begin{aligned} \Delta r(k) &= r(k) - r(k-1) \\ \Delta y(k) &= y(k) - y(k-1) \end{aligned} \tag{8}$$

At each time step, the error is modified through gradient descent method with(5):

$$e_k = [u(k-\Delta) - \tilde{u}(k)] \tag{9}$$

The weights of the fuzzy inverse model are copied for the fuzzy controller. To minimize the error $e(k)$, the weights corresponding to the rules activated for the inverse model will be adjusted. In the meantime the weights corresponding to fuzzy controller will remain unchanged. After the adjusting time, the weight of rules of inverse model is copied to the fuzzy controller.

$$y(k)_{\text{plant output}} \cong r(k-\Delta)_{\text{fuzzy controller input}} \tag{10}$$

5. CONTROL of a BALL and BEAM SYSTEM

5.1 Ball and Beam System

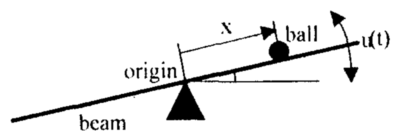

Fig. 5. The Ball and Beam system.

We use the gradient descent method to approximate the fuzzy inverse model for nonlinear ball and beam system. Assuming that dynamics of plant are unknown, we have implemented the fuzzy controller with an IBM compatible PC. The ball and beam system is shown in Fig. 5. The beam is made to rotate in a vertical plane by applying a torque at the center of rotation and the ball is free to roll along the beam.

From (Seo, Y. R. 1994), the system is represented following nonlinear equations.

$$\ddot{x} = \frac{\begin{pmatrix} 1.69571640\dot{\theta}x - 16.61802072x\sin\theta + 0.001591819\tau \\ -0.000213303\dot{\theta}x\dot{x} - 0.001045188x\cos\theta + 0.000194043\sin\theta \\ +0.0000077784\dot{\theta}^2 x - 0.00007628 4x\sin\theta \end{pmatrix}}{(0.00118825 + 0.00264726x^2)} \tag{11}$$

$$\ddot{\theta} = \frac{\begin{pmatrix} 0.0395114\,\tau - 0.0052945270\dot{\theta}x\dot{x} - 0.025943185x\cos\theta \\ +0.004816044\sin\theta + 0.000193213\dot{\theta}x - 0.001893489x\sin\theta \end{pmatrix}}{(0.00118825 + 0.00264726x^2)} \quad (12)$$

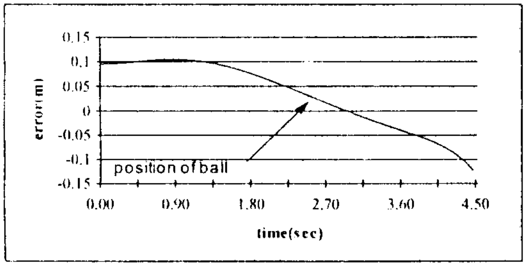

Fig. 6. Error of ball (initial position: 0.095m, goal: 0.0m)

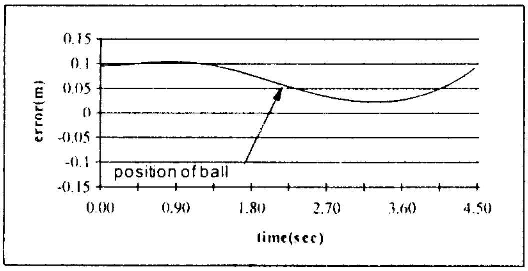

Fig. 7. Error of ball (initial position: 0.095m, goal: 0.0m)

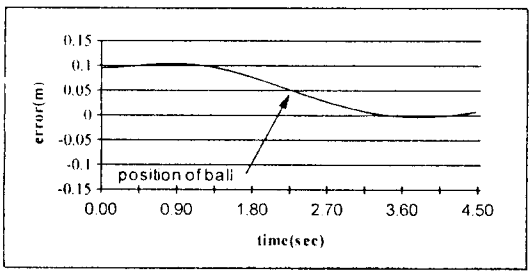

Fig. 8. Error of ball (initial position: 0.095m, goal: 0.0m)

6. CONCLUSIONS and FUTURE WORK

The validity of the self-learning fuzzy controller with a fuzzy approximation of the inverse modeling can be evaluated through the computer simulation of the plant. In Fig. 6, and Fig. 7, we can obtain the results of simulation which are insufficient learning states. But Fig. 8 shows a good learning state with small steady-state error, the position of ball is almost up to the goal which is point zero.

Using the fuzzy inverse modeling and gradient descent method instead of a complex classical method, we can get the goal that makes the design of self-learning fuzzy controller less complex.

And the fuzzy controller is able to learn proper action for unknown nonlinear system control. Now, we have implemented the fuzzy controller("The Balance Boy") with a 386PC, an interface board, a D/A converter, a PWM motor drive-circuit, a counter, and a sensing-circuit for ball position.

For better performance of learning, the fast convergence learning method or the hybrid learning method will be needed. The hybrid learning method which is composed as the least mean square error method and the gradient descent method.

7. REFERENCES

Narendra, K.S. and K. Parthasarathy(1990) Identification and control of dynamical systems using neural networks, IEEE Trans. on Neural Network, Vol.1, pp.4-27

Wang, L.X. and J.M. Mendel (1992) Gradient descent fuzzy system as nonlinear dynamic system identifiers, IEEE int. Conf. Fuzzy System, pp.1409-1418

Widrow, B. and S.D. Stearns(1985) Adaptive signal processing, pp.231-224, Prentice-Hall, New Jersey

Wang, L(1992) Fuzzy System are Universal Approximators, IEEE int. Conf. Fuzzy System, pp.1163-1168.

Lee, C.C.(1990) Fuzzy logic in control systems: fuzzy logic controller, part I, IEEE Trans., on Syst., Man, and Cybern., Vol. SMC-20, pp.404-418

Lee, C.C.(1990) Fuzzy logic in control systems: fuzzy logic controller, part II, IEEE Trans., on Syst., Man, and Cybern., Vol. SMC-20, pp.419-435

Zadeh, L. A.(1965) Fuzzy sets, Informat. Control, Vol. 8, pp.338-353

Seo, E. T. and C. H. Chung(1994) Fuzzy Systems as Nonlinear Adaptive Filters, Kwangwoon Univ., ICIS-TR-RAL-946

Widrow, B. and S.D. Stearns(1985) Adaptive signal processing, pp.99-109, Prentice-Hall, New Jersey

Seo, Y. R. and C. H. Chung(1994) Design of a Fuzzy Controller with IBM-386PC and Its application to a Ball & Beam System, Kwangwoon Univ., ICIS-TR-RAL- 941

Adaptive Sliding-Mode Tracking Control in the Presence of Unmodeled Dynamics

Seung Ho Cho, *Associate Professor*
Department of Mechanical Engineering
Hong-Ik University
Mapo-Ku, Seoul 121-791, KOREA

Abstract

To increase the robustness of the feedforward tracking control system, a new discrete time sliding function has been defined and utilized for the formulation of control law. In adaptive case the robustness is achieved by using both a normalized gradient algorithm with deadzone and a sliding function-based nonlinear feedback, while in nonadaptive case by using only a sliding function -based nonlinear feedback.

1 Introduction

When the desired position is time varying,the tracking performance can be significantly improved by the feedforward tracking controller. For minimum phase systems the feedforward tracking controller can be designed to achieve perfect tracking based on stable pole-zero cancellation.

In fact the model/plant mismatches eventually lead to an imperfect pole-zero cancellation, which in turn may significantly degrade the tracking performance and even result in the instabilities of the overall system. Mismatch in pole-zero cancellation causes the motivation of using Diophantine equation for constructing sliding function.

In analyzing these uncertainties we adopt the following two cases,i.e.,nonadaptive and adaptive. In nonadaptive case we construct the nonlinear feedback control law which is based on discrete time sliding function. In adaptive case these uncertainties may cause the divergence of the adaptive process, which has been resulted in the introduction of deadzone in the estimator to bring the robust results in the adaptive control system(*Egardt 1978*,*Samson 1983*,*Ioannou and Kokotovic 1983*,*Kreisselmeier and Anderson 1986*).

Deadzone and sliding boundary layer is intended to provide robustness in control system against modeling errors. In this paper we try that their design concept be set up on the same basis, and try to match some relationship between them.

2 Robust Discrete Time Tracking Control

The controlled plant is assumed to be represented by the following discrete time model:

$$y(k) = \frac{z^{-d}\, B(z^{-1})}{A(z^{-1})}\, u(k) \quad + \eta(k) \tag{1}$$

where

- $u(k)$ and $y(k)$ are the measurable input and output respectively,
- $\eta(k)$ represents the modeling error,
- $A(z^{-1})$ and $B(z^{-1})$ are polynomials of order n and m respectively in the backward shift operator z^{-1}.

The order n and m as well as the delay step d are assumed to be known. It is further assumed that
A1 : $A(z^{-1})$ and $B(z^{-1})$ are coprime.
A2 : $|\ \eta(k)\ | \leq \mu\ m(k)$, where μ is a positive scalar and $m(k)$ is defined by

$$m(k) = \sigma\ m(k-1) + |\ u(k-1)\ | + |\ y(k-1)\ |. \tag{2}$$

with $0 < \sigma < 1$.
A3 : $\|\theta_1\| \leq \rho_1$, where $\theta_1^T = [a_1, \cdots, a_n, b_0, \cdots, b_m]$ and ρ_1 is a known positive scalar.

The plant (1) can also be expressed as

$$y(k) = \theta_1^T\ \phi_1(k) + \eta_1(k) \tag{3}$$

where

$$\begin{aligned} \phi_1^T(k) &= [-y(k-1), \cdots, -y(k-n), u(k-d), \\ &\quad \cdots, u(k-d-m)], \\ \eta_1(k) &= A(z^{-1})\eta(k). \end{aligned}$$

Then it follows from A2 ~ A3 and (2) that

$$|\ \eta_1(k)\ | = |\ A(z^{-1})\eta(k)\ | \leq \nu_1\ \mu m(k) \tag{4}$$

$$\nu_1 = 1 + \rho_1 \sigma^{-n} n \quad (5)$$

In this paper we propose a robust discrete time tracking control by combining pole/zero cancellation and discrete time version of sliding control. For assignment of closed loop poles and setting the structure of the control system, we utilize the following Diophantine equation.

$$D_1(z^{-1}) = A(z^{-1})S_1(z^{-1}) + z^{-d}R_1(z^{-1}) \quad (6)$$

where

From equations (6) and (4), the following equation is derived.

$$D_1(z^{-1})y(k) = \theta_2^T\,\phi_2(k-d) + \eta_2(k) \quad (7)$$

where $R_1(z^{-1})$, $S_1(z^{-1})$ and $D_1(z^{-1})$ are polynomials of order n-1,d-1 and n respectively in the backward shift operator z^{-1}.

$$\theta_2^T = [b_0,\cdots,b_m s_{d-1}, r_0, r_1,\cdots,r_{n-1}], \quad (8)$$

$$\phi_2^T(k) = [u(k), u(k-1),\cdots,u(k-m-d+1), y(k), y(k-1),\cdots,y(k-n+1)]. \quad (9)$$

and $\eta_2(k)$ is expressed as follows

$$\eta_2(k) = \theta_\eta^T\,\phi_\eta(k) \quad (10)$$

To derive the bounds of $|\theta_2^T\phi_2(k-d)|$ and $|\eta_2(k)|$ it is necessary to have the following assumption.

A4: $\|\theta_d\| < \rho_2$, where $\theta_d^T = [s_1,\cdots,s_{d-1}, r_0,\cdots,r_{n-1}]$ and ρ_2 is a known positive scalar.

Proposition 1 $\quad |\theta_2^T\,\phi_2(k-d)| < K_{m1}\,\sigma^{1-d-r}m(k)$,

where $r = max\ deg(n, m+d)$.

$$K_{m1} = (m+1)\rho_1 + (m+1)(d-1)\rho_1\rho_2 + n\rho_2.$$

It follows with assumption A4 that

$$|\eta_2(k)| = |A(z^{-1})\,S_1(z^{-1})\,\eta(k)| \le \nu_2\mu\, m(k) \quad (11)$$

with

$$\nu_2 = 1 + 3\rho_1\rho_2 n(d-1)\,\sigma^{-n-(d-1)} \quad (12)$$

In designing a robust discrete time tracking controller we define $s(k)$ by

$$s(k) = D_1(z^{-1})[y(k) - y_m(k)] \quad (13)$$

and add a control loop with $s(k)$ to compensate the modeling error $\eta_2(k)$. Outside the boundary layer ($|s(k)| > \Phi$) as well as inside the boundary layer ($|s(k)| \le \Phi$) the control law becomes

$$u(k) = \frac{1}{b_0}\left[s(k) + D_1(z^{-1})y_m(k+d) - \theta_2^T\phi_2(k) - K\,sat\left\{\frac{s(k)}{\Phi}\right\}\right] \quad (14)$$

where

$$sat\left\{\frac{s(k)}{\Phi}\right\} = \begin{cases} +1, & \text{for } \Phi < s(k), \\ \frac{s(k)}{\Phi}, & \text{for } -\Phi < s(k) < \Phi, \\ -1, & \text{for } s(k) < -\Phi. \end{cases} \quad (15)$$

and θ_2 and $\phi_2(k)$ are defined by

$$\theta_2^T = [b_0, \theta_2'^T] \quad (16)$$

$$\phi_2^T(k) = [u(k), \phi_2'^T(k)] \quad (17)$$

Proposition 2 *For arbitrary* $d_0 > 0$ *there exists a* $\mu_0 > 0$ *(which depends on* d_0*) such that for all* $0 \le \mu \le \mu_0$ *and arbitrary initial conditions, it follows that*

$$|s(k)| < d_0\, m(k)$$

where $d_0 = \nu_2\mu_0$.

Proposition 3 *$m(k)$ is bounded, in turn $u(k)$ and $y(k)$ are bounded.*

The condition that Φ and K satisfy will be explained in theorem 1.

Notice that from Eqs (13) and (14), $s(k)$ satisfies

$$s(k+d) = s(k) + \eta_2(k+d) - K\,sat\left\{\frac{s(k)}{\Phi}\right\} \quad (18)$$

The robust stability of the robust discrete time tracking control system is proved in the following theorem.

Theorem 1 *Robust Stability of the Robust Discrete Time Tracking Control System :*

The robust discrete time tracking control system, consisting of the plant (1) and the control law (14), is stable in the sense that $|s(k)|$ decreases when $|s(k)| > \Phi$ and that the steady state value of s is bounded by Φ.

Proof: Introduce the following discrete Lyapunov function candidate

$$V(k) = |s(k)| \quad (19)$$

which can be interpreted as the distance to the surface $s(k) = 0$.

Then we can formulate the followig difference equation.

$$\begin{aligned} \Delta V(k+d) &= V(k+d) - V(k) \\ &= |s(k+d)| - |s(k)| \\ &= |s(k) + \eta_2(k+d) - K\,sgn\{s(k)\}| \\ &\quad - |s(k)| \end{aligned} \quad (20)$$

Based on the Proposition 3 let us assume that the upperbound of modeling error is constant, i.e.

$$|\eta_2(k)| \le \nu_2\mu\, m(k) < F_2 \quad (21)$$

where F_2 is a bounded scalar. Then the following a couple of condition on K make (20) negative.

$$K > F_2,\ K < 2\,|s_i(k)| - F_2 \quad (22)$$

If K is selected as following

$$K = F_2 + \eta_0 \quad (23)$$

Then (19) will be a Lyapunov function in the following region.

$$|s_i(k)| > F_2 + \frac{\eta_0}{2} \tag{24}$$

Inside the boundary layer, the s-dynamics become :

$$s(k+d) = (1-\frac{K}{\Phi})s(k) + \eta_2(k+d) \tag{25}$$

The boundary layer thickness Φ can be selected such that (25) shows characteristics of a first-order filter with input $\eta_2(k+d)$ and eigenvalue.

$$1 - \frac{K}{\Phi} = \lambda \tag{26}$$

From (23) and (26):

$$\Phi = \frac{F_2 + \eta_0}{1-\lambda} \tag{27}$$

From the stability viewpoint the following relation should be satisfied.

$$|1 - \frac{K}{\Phi}| < 1 \tag{28}$$

For a stable eigenvalue λ the steady state solution becomes:

$$\lim_{k\to\infty} s_i(k) = \lim_{k\to\infty}\sum_{j=0}^{\infty}\lambda^j\,\eta_2(k-j\cdot d) \tag{29}$$

From (29) the following relations are derived.

$$|\lim_{k\to\infty} s(k)| \le \left\{1+|\lambda|+|\lambda|^2+\cdots\right\}F_2 \tag{30}$$

$$|\lim_{k\to\infty} s(k)| < \frac{1}{1-|\lambda|}F_2 < \frac{K}{1-|\lambda|} \tag{31}$$

$$|\lim_{k\to\infty} s(k)| < \Phi \tag{32}$$

3 Robust Adaptive Discrete Time Tracking Control

To maintain the control objective for a plant with unknown dynamics due to modeling error in real time, parameter adaptation schemes coupled with the control law needs to be investigated. A standard approach to incorporate the adaptation scheme with the control law in stability analysis is to guarantee that a norm of the difference between estimated parameters and actual parameters decreases at each step. Deadzone concept has been utilized in most robust adaptive schemes. In this paper we apply normalized gradient parameter estimator as our adaptation scheme.

The algorithm is given by:

$$\hat\theta_2(k) = \hat\theta_2(k-1) + \frac{a(k)\phi_2(k-d)D(e(k),d(k))}{\phi_2^T(k-d)\phi_2(k-d)+c} \tag{33}$$

where $k \ge d$, $0 < a(k) < 2$, $c > 0$, and $D(e(k),d(k))$ is the continuous function defined by

$$D(e(k),d(k)) = \begin{cases} e(k) - d(k)sgn\{e(k)\}, & \text{if } |e(k)| \ge d(k). \\ 0, & \text{otherwise.}\end{cases} \tag{34}$$

and :

$$\begin{aligned} e(k) &= \theta_2^T\phi_2(k-d) + \eta_2(k) - \hat\theta_2^T(k-1)\phi_2(k-d) \\ &= \tilde\theta_2^T(k-1)\phi_2(k-d) + \eta_2(k) \end{aligned} \tag{35}$$

Proposition 4 *The estimation algorithm defined by (33) – (35), when applied to plant (7), has the following properties:*

$$\lim_{k\to\infty}\|\hat\theta_2(k) - \hat\theta_2(k-1)\| = 0 \tag{36}$$

$$\lim_{k\to\infty} D(e(k),d(k)) = 0 \tag{37}$$

In the notation of this section the control law (14) becomes:

$$u(k) = \frac{1}{\hat b_0}[s(k) + D_1(z^{-1})y_m(k+d) - \hat{\bar\theta}_2^T\bar\phi_2(k) + K\,sat\{\frac{s(k)}{\Phi}\}] \tag{38}$$

A reasonable criteria to use in shifting from the non-adaptive to adaptive controller is whether s is inside or outside the boundary layer. That is, when inside the boundary layer, the control law of (38) is used and adaptation does not occur. The control laws should be continuous at the boundary layer edge. For this purpose, note that the non-adaptive control law of (38) takes on the following value at the boundary layer edge.

$$u(k) = \frac{1}{\hat b_0}[D_1(z^{-1})y_m(k+d) - \hat{\bar\theta}_2^T\bar\phi_2(k) + \lambda\Phi sgn\{s(k)\}] \tag{39}$$

The control law of (39) reduces to :

$$\begin{aligned} D_1(z^{-1})y_m(k+d) &= -\lambda\Phi\, sgn\{s(k)\} + \hat b_0 u(k) + \hat{\bar\theta}_2\bar\phi_2(k) \\ &= -\lambda\Phi\, sgn\{s(k)\} + \hat\theta_2\phi_2(k) \end{aligned} \tag{40}$$

Recalling (7) :

$$D_1(z^{-1})y(k+d) = \theta_2\phi_2(k) + \eta_2(k+d) \tag{41}$$

From (40) and (41) the s-dynamics becomes :

$$s(k+d) = \lambda\Phi\, sgn\{s(k)\} + \tilde\theta_2\phi_2(k) + \eta_2(k+d) \tag{42}$$

Combining (42) and estimation algorithm yields :

$$\lim_{k\to\infty}\frac{[s(k) - \lambda\Phi sgn\{s(k-d)\} - d(k)sgn\{e(k)\}]^2}{\phi_2^T(k-d)\phi_2(k-d)+c} = 0 \tag{43}$$

$$e(k) = s(k) - \lambda\Phi\, sgn\{s(k-d)\} \tag{44}$$

Now select $d(k)$ as :

$$d(k) = (1-\lambda)\Phi \tag{45}$$

From (27), this guarantees that $d(k) \ge F_2$ as required. The condition on adaptation remains as $|e(k)| \ge d(k)$, which from (44) and (45) is met if $|s(k)| > \Phi$ (being outside the boundary layer). With this condition satisfied, from (44) it can be seen that $sgn\{e(k)\} = sgn\{s(k)\}$, and so using (45), (43) becomes :

$$\lim_{k\to\infty}\frac{[s(k) - \Phi\, sgn\{s(k-d)\}]^2}{\phi_2^T(k-d)\phi_2(k-d)+c} = 0 \tag{46}$$

Note that condition (1) of Key Technical Lemma[*Goodwin and Sin* 1984) is satisfied, with $p(k) = s(k) - \Phi\, sgn\{s(k)\}$, $b_1(k) = c$, $b_2(k) = 1$, and $\sigma(k) = \phi_2(k-d)$. This choice of $b_1(k)$ and $b_2(k)$ also satisfies condition (2). To show that condition (3) of the lemma is satisfied, it is necessary that $|s(k) - d(k) sgn\{s(k-d)\}|$ linearly bounds $\phi(k-d)$. This can be done by first showing that $y(k)$ linearly bounds $\phi(k-d)$. Notice that $\phi(k-d)$ is made up of functions of the output $y(i)$, $k-n+1 \le i \le k$ and of the control input $u(i), k-m-d+1 \le i \le k$. Since the parameter estimates remain bounded due to the boundedness properties of the parameter estimator, $y(i)$ and $u(i)$ are also bounded as it was demonstrated by Proposition 3. To demonstrate that $|s(k) - \Phi\, sgn\{s(k-d)\}|$ linearly bounds $y(k)$, recall that the selection of function $s(k)$ in Section 2 established the ability to set the dynamics on the sliding surface as desired. Since $s(k)$ is bounded and $y_m(k)$ is assumed to be bounded, $s(k)$ can linearly bound $y(k)$. So $|s(k) - \Phi\, sgn\{s(k-d)\}|$ linearly bounds $y(k)$. With all conditions of the Key Technical Lemma satisfied, this allows for the results that $\phi(k-d)$ is bounded for all k and that

$$\lim_{k \to \infty} |s(k) - \Phi\, sgn\,\{s(k-d)\}| = 0.$$

The results thus proved is summarized in the following theorem.

Theorem 2 *Robust Stability of the Robust Adaptive Discrete Time Tracking Control System :*

The robust adaptive discrete time tracking control system, consisting of the plant (1),*the controller* (38),*and the adaptive law* (33) – (35) *satisfying* (45), *is stable in the sense that* $| s(k) |$ *decreases when* $| s(k) | > \Phi$ *and that the steady state value of* s *is bounded by* Φ.

Both in adaptive and in nonadaptive case we have arrived at the robust same results.

4 Conclusion

Based on the Diophantine equation a new discrete time sliding function has been defined and utilized for the robust feedforward tracking control law. The sliding boundary layer is composed of eigenvalue in s-dynamics and of upperbound in modeling error. The modeling error which ultimately affects on the tracking performance is diluted in the sliding boundary layer, which results in increase of robustness. Parameter estimator with a new deadzone is proposed to robustly estimate the unknown parameters of plant. Using eigenvalue as an intermediate variable,the relationship between deadzone and sliding boundary layer is derived to provide compatibility in robust control parameters. In adaptive case the robustness is achieved by using both a normalized gradient algorithm with deadzone and a sliding function-based nonlinear feedback, while in nonadaptive case by using only a sliding function -based nonlinear feedback.

References

[1] Egardt,B.,1979,"Stability of Adaptive Controllers (Lecture Notes in Control and Information Sciences)",New York: Springer-Verlag.

[2] Samson,C.,1983,"Stability Analysis of Adaptively Controlled Systems Subject to Bounded Disturbances", Automatica,vol.19,81-86.

[3] Ioannou, P.and Kokotovic,P.,1983,"Adaptive Systems with reduced models", Springer Series,Lect.Notes in Control Inf.Sci.47.

[4] Kreisselmeier,G. and Anderson,B.D.O.,1986,"Robust Model Reference Adaptive Control",IEEE Trans.Automatic Control,vol.AC-31,no.2,127-133.

[5] Goodwin,G.C. and Sin,K.S.,1984,"Adaptive Filtering Prediction and Control", Englewood Cliffs,NJ:Prentice Hall.

[6] Ortega,R.and Lozano,R.,1987,"A Note on Direct Adaptive Control of System with Bounded Disturbances", Automatica,253-254.

[7] Landau,I.D. and Lozano,R.,1981,"Unification and Evaluation of Discrete Time Explicit Model Reference Adaptive Design",Automatica,vol.17,no.4, 593-611.

[8] Anderson,B.D.O. and J.B.Moore,J.B.,1971,Linear Optimal Control, Prentice Hall.

[9] Tomizuka,M. and Whitney,D.E.,1975 "Optimal Discrete Finite Preview Problem(Why and How is Future Information Important ?)",ASME Journal of Dynamic Systems,Measurement and Control,vol.97,no.4,319-325.

[10] Utkin,V.I.,1977,"Variable Structure Systems with Sliding Modes",IEEE Trans.Automatic Control,vol.AC-22, no.2,212-222.