Model free adaptive control with pseudo partial derivative based on fuzzy rule emulated network.

Chidentree Treesatayapun[1]

[1]Department of Robotic and Advanced Manufactory, CINVESTAV-Saltillo, Ramos Arizpe, 25900, Mexico

Abstract— *In this article, a model free adaptive control with the estimated pseudo partial derivative (PPD) is introduced by multi-input fuzzy rules emulated network (MiFREN) for a class of discrete-time systems. The resetting mechanism can be relaxed in this adaptation scheme. Human knowledge about the controlled plant is rearranged to define the IF-THEN rules directly. Those fixed parameters are designed according to guarantee the convergence related on the controller performance. All adjustable parameters inside MiFREN are tuned by the proposed on-line learning algorithm. The design example and simulation results demonstrate the performance of the proposed controller under the nominal system and system with disturbances.*

Keywords: Fuzzy logic; Model-free adaptive controller; Pseudo-partial derivative; Discrete-time

1. Introduction

Model free adaptive control (MFAC) has been continuously developed for several systems with unknown or ill-defined model especially for a class of discrete-time domain [1], [2]. With out any mathematic model of the plant, a linearization concept based on pseudo-partial derivative (PPD) is composed as the equivalent system. Due to the comparison with other adaptive controllers, only real time measured data of the controlled plant is necessary to establish MFAC. Unlike model reference adaptive control based on neural networks, the off-line tuning phase can be excluded because the time-varying PPD can be tuned by real time measurement data only [3].

In general, The control law has been determined by PPD under some necessary constrains and the resetting technique. It's clear that the control system performance depends on the accuracy of PPD estimation. The compensation system based on an artificial neural network has been discussed in [4] by the additional of another control effort. By using the on-off controller [5], the system dynamic can be considered as the unknown system when the switching sequence can be adapted by a feedback controller.

According to the controlled plant, in practice, PPD seems like the system sensitivity which can be understood by human knowledge related on input-output behavior. In this work, this knowledge can be reformulated into IF-THEN rules for an adaptive network called multi-input fuzzy rule emulated network(MiFREN) [6]. The on-line adaptation is developed to tune all adjustable parameters inside MiFREN. Thus, the time variation of PPD can be directly estimated by MiFREN according to our proposed cost function. Moreover, the resetting algorithm, which is needed for tuning PPD [7], can be relaxed in this work because of MiFREN's property [8] which is directly related on the change of input respectively to the plant's output. All fixed parameters are designed to guarantee the convergence which can affect the performance of controller.

This paper is organized as the follows. Section 2 introduces the problem formulation and some details about the dynamic linearization equivalent model. The control law is proposed with the estimated PPD based on MiFREN in section 3. Section 4 presents the MiFREN configuration together with the parameters adaptation. The design example and simulation results are demonstrated in section 5 including the consideration of disturbance effect. Section 6 represents our conclusions.

2. Problem Formulation

In this work, the nonlinear system for a class of discrete-time domain can be described by

$$y(k+1) = f(y(k), \cdots, y(k-n_y), u(k), \cdots, u(k-n_u)), \tag{1}$$

where $y(k) \in \mathbb{R}$ and $u(k) \in \mathbb{R}$ denote as the time index k system output and input, respectively with the unknown orders n_y and n_u. The nonlinear function $f(\cdot)$ is definitely unknown. According to the conventional MFAC algorithms, those following assumptions are stated.

Assumption 1: The partial derivatives of $f(\cdot)$ are continuous with respect to the control effort $u(k)$.

Assumption 2: The nonlinear system described in (1) is generalized Lipschitz. That means the positive constant l must be defined when $|\Delta y(k+1)| \leq l|\Delta u(k)|$, when $\Delta y(k+1) = y(k+1) - y(k)$ and $\Delta u(k) = u(k) - u(k-1)$.

According to those upper assumptions, the following lemma can be obtained.

Lemma 1: The nonlinear system (1) satisfied assumption 1 and 2 with $|\Delta u(k)| \neq 0$ for time index k, can be transformed into the equivalent compact form dynamic linearization (CFDL) as

$$\Delta y(k+1) = \Phi(k)\Delta u(k), \tag{2}$$

when $\Phi(k)$ denotes the pseudo partial derivative (PPD).
Proof: The proof is omitted here. Reader can refer to [3] for more details and complete proof.

□

The parameter PPD can be estimated by several learning algorithm such as the modified projection. On the other hand, in this work, PPD is determined by the adaptation network MiFREN which can be operated under the human knowledge according to the controlled plant.

3. Control law

In this section, the control effort $u(k)$ will be formulated. Let define the cost function as

$$J(u(k)) = [r(k+1) - y(k+1)]^2 + \lambda[u(k) - u(k-1)]^2, \quad (3)$$

when λ is a designed weighting constant. By using (2), the cost function can be rewritten as

$$J(u(k)) = [r(k+1) - y(k) - \Phi(k)\Delta u(k)]^2 + \lambda[\Delta u(k)]^2. \quad (4)$$

Differentiating (4) with respect to $u(k)$, we have

$$\begin{aligned}\frac{\mathrm{d}J(u(k))}{\mathrm{d}u(k)} &= -2[r(k+1) - y(k)]\Phi(k) + 2\Phi^2(k)\Delta u(k) \\ &\quad +2\lambda\Delta u(k). \quad (5)\end{aligned}$$

Let $\frac{\mathrm{d}J(u(k))}{\mathrm{d}u(k)} = 0$, thus

$$\Delta u(k) = \frac{\Phi(k)[r(k+1) - y(k)]}{\lambda + \Phi^2(k)}. \quad (6)$$

The idea control law can be obtained as

$$u(k) = u(k-1) + \frac{\Phi(k)[r(k+1) - y(k)]}{\lambda + \Phi^2(k)}. \quad (7)$$

In practice, the output $y(k)$ can be measured but $\Phi(k)$ cannot be directly determined because of the unknown nonlinear system $f(\cdot)$. To realize this control law, $\Phi(k)$ will be estimated by MiFREN as $\hat{\Phi}(k)$, thus, the control effort can be generated by

$$u(k) = u(k-1) + \rho\frac{\hat{\Phi}(k)[r(k+1) - y(k)]}{\lambda + \hat{\Phi}^2(k)}, \quad (8)$$

when ρ is a step-size constant. Both ρ and λ are designed parameters which can be given by the following lemma to guarantee the convergence of $u(k)$. Before the proof of control effort boundary, the follow assumption is necessary.
Assumption 3: The direction of $\hat{\Phi}(k)$ or sign$\left\{\hat{\Phi}(k)\right\}$ is unknown but it doesn't change such as sign$\left\{\hat{\Phi}(k)\right\}$ = sign$\left\{\hat{\Phi}(k-1)\right\}$.
Remark: This assumption can be existed in several practical systems such as the chemical process, robotic system and so on. Moreover, this assumption will be proved again by the experimental results next.

Lemma 2: According to assumption 2 and the MiFREN's property, the maximum of $\hat{\Phi}(k)$ must be existed as $\hat{\Phi}_M, \forall k \in \mathrm{N}$ and both parameters ρ and λ are designed by following this relation

$$0 < \rho < \frac{2\lambda}{\hat{\Phi}_M}, \quad (9)$$

thus, the control effort from (8) is bounded.
Proof: Substitute $y(k) = \hat{\Phi}(k-1)[u(k-1) - u(k-2)] - y(k-1)$ into (8), we have

$$\begin{aligned}u(k) &= u(k-1) - \frac{\rho\hat{\Phi}(k)\hat{\Phi}(k-1)}{\lambda + \hat{\Phi}^2(k)}u(k-1) \\ &\quad + \frac{\rho\hat{\Phi}(k)}{\lambda + \hat{\Phi}^2(k)}\Big[r(k+1) + \hat{\Phi}(k-1)u(k-2) \\ &\quad + y(k-1)\Big], \\ &= A_u(k)u(k-1) + \xi_u(k), \quad (10)\end{aligned}$$

where

$$A_u(k) = 1 - \frac{\rho\hat{\Phi}(k)\hat{\Phi}(k-1)}{\lambda + \hat{\Phi}^2(k)}, \quad (11)$$

and

$$\xi_u(k) = \frac{\rho\hat{\Phi}(k)}{\lambda + \hat{\Phi}^2(k)}\Big[r(k+1) + \hat{\Phi}(k-1)u(k-2) + y(k-1)\Big]. \quad (12)$$

Substitute (9) into (13) and use assumption 3, it is clear that

$$A_u(k) < 1. \quad (13)$$

□

4. MiFREN for pseudo partial derivative

4.1 PPD based on MiFREN

MiFREN is an adaptive network which can be operated by human knowledge in the format of IF-THEN rules. The design of IF-THEN rules is also a key of the performance. The knowledge based on the controlled plant is roughly necessary. Due to (2), we can rearrange the relation of $\hat{\Phi}(k)$ as

$$\hat{\Phi}(k) \doteq \frac{\Delta y(k)}{\Delta u(k-1)}, \quad (14)$$

when $\Delta u(k) \neq 0$.

According to the upper relation, we can define some example IF-THEN rules as the followings:

- IF $\Delta y(k)$ is Positive Large and $\Delta u(k-1)$ is Positive Large THEN $\hat{\Phi}(k)$ should be Positive small,
- IF $\Delta y(k)$ is Positive Large and $\Delta u(k-1)$ is Positive small THEN $\hat{\Phi}(k)$ should be Positive Large,
- $\ldots$,
- IF $\Delta y(k)$ is Negative Large and $\Delta u(k-1)$ is Positive Large THEN $\hat{\Phi}(k)$ should be Negative small,
- $\ldots$,

- IF $\Delta y(k)$ is Negative Large and $\Delta u(k-1)$ is Negative small THEN $\hat{\Phi}(k)$ should be Positive Large,
- IF $\Delta y(k)$ is Negative Large and $\Delta u(k-1)$ is Negative Large THEN $\hat{\Phi}(k)$ should be Positive small.

With this implementation, unlike the conventional algorithms to determine $\hat{\Phi}(k)$, the sign of $\hat{\Phi}(k)$ is directly determined by those IF-THEN rules. To simplify, those IF-THEN rules can be defined by Table 1.

$\Delta y(k)$					
PL	PS	AZ	NS	NL	$\Delta u(k-1)$
P_1	P_6	C_{11}	N_{16}	N_{21}	PL
P_2	P_7	C_{12}	N_{17}	N_{22}	PS
ε_3	ε_8	ε_{13}	ε_{18}	ε_{23}	AZ
N_4	N_9	C_{14}	P_{19}	P_{24}	NS
N_5	N_{10}	C_{15}	P_{20}	P_{25}	NL

Table 1: IF-THEN rules

The linguistic variables PL, PS, AZ, NS and NL denote as Positive Large, Positive Small, Almost Zero, Negative Small and Negative Large, respectively. In this case, the number of IF-THEN rules is 25 and the relation the THEN-part can be given by other linguistic variable as $\square_i$ for $i = 1, 2, \cdots, 25$. The variables P, C and N stand for Positive, Close to zero and Negative, respectively. Unlike the conventional technique, when Δu nearly reaches to zero or stays in the range to AZ-membership function, the designed parameter ε is defined as a small positive constant and $\varepsilon_3 = \varepsilon_8 = \varepsilon_{13} = \varepsilon_{14} = \varepsilon_{23} = \varepsilon$. Moreover, the parameters ε_i can be given by difference values with the on-line learning algorithm which will be discussed next.

According to those IF-THEN rule, the network architecture of MiFREN can be constructed in Fig.1. The estimated

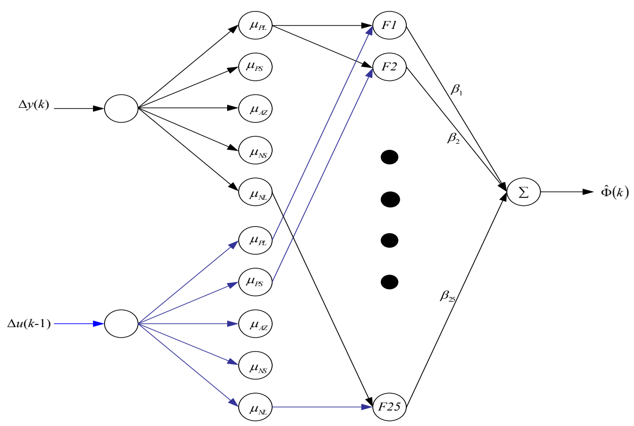

Figure 1: MiFREN networking structure.

PPD or $\hat{\Phi}(k)$ can be obtained by

$$\hat{\Phi}(k) = \sum_{i=1}^{25} F_i(\Delta y(k), \Delta u(k-1))\beta_i, \tag{15}$$

when β_i is a Linear Consequence (LC) parameter for THEN-part linguistic variable $\square_i$. The function $F_i(\cdot)$ is the i^{th} rule relation determined by membership functions related on the i^{th} rule. For example, according to Table 1, we have $F_1(k) = \mu_{PL}(\Delta y(k))\mu_{PL}(\Delta u(k-1))$ at rule # 1, $F_{16}(k) = \mu_{NS}(\Delta y(k))\mu_{PL}(\Delta u(k-1))$ at rule # 16 and $F_{25}(k) = \mu_{NL}(\Delta y(k))\mu_{NL}(\Delta u(k-1))$ at rule # 25. In this work, we can set those membership functions and LC parameters or β_i by the human knowledge based on the controlled plant. More explanations with example will be discussed in the next section.

4.2 Parameters adaptation

The on-line learning algorithm will be applied in this work for LC parameters. Thus, the estimated PPD in (15) can be rewritten by

$$\hat{\Phi}(k) = \sum_{i=1}^{25} F_i(k)\beta_i(k), \tag{16}$$

when $\beta_i(k)$ denotes the LC parameter for the i^{th} rule at time index k. To tune this parameter, the cost function $J(\hat{\Phi}(k))$ can be given as the following:

$$\begin{aligned} J(\hat{\Phi}(k)) &= \frac{1}{2}[\Delta y(k) - \hat{\Phi}(k)\Delta u(k-1)]^2 \\ &\quad + \frac{\gamma}{2}[\hat{\Phi}(k) - \hat{\Phi}(k-1)]^2, \end{aligned} \tag{17}$$

when γ is a positive defined constant. Determine the derivative with respect to $\beta_i(k)$, thus, we obtain

$$\begin{aligned} \frac{\partial J(\hat{\Phi}(k))}{\partial \beta_i(k)} &= \frac{\partial J(\hat{\Phi}(k))}{\partial \hat{\Phi}(k)} \frac{\partial \hat{\Phi}(k)}{\partial \beta_i(k)} \\ &= \Big[[\gamma + \Delta u^2(k-1)]\hat{\Phi}(k) - \gamma\hat{\Phi}(k-1) \\ &\quad -\Delta y(k)\Delta u(k-1)\Big]F_i(k). \end{aligned} \tag{18}$$

According to gradient search, the tuning law can be defined by

$$\beta_i(k+1) = \beta_i(k) - \eta_\beta \frac{\partial J(\hat{\Phi}(k))}{\partial \beta_i(k)}, \tag{19}$$

where η_β is the learning rate which can be designed as a small positive constant. By using (19) and (18), the tuning law can be formulated as

$$\begin{aligned} \beta_i(k+1) &= \beta_i(k) - \eta_\beta\Big[[\gamma + \Delta u^2(k-1)]\hat{\Phi}(k) \\ &\quad -\gamma\hat{\Phi}(k-1) - \Delta y(k)\Delta u(k-1)\Big]F_i(k). \end{aligned} \tag{20}$$

The designed parameters γ and η_β can effect to the convergence of β. The following lemma introduces the relation which guarantees the convergence of tuned parameters β_i.

Lemma 3: The LC parameters β_i for $i = 1, 2, \cdots, N_F$, when N_F denotes the number of IF-THEN rules, are bounded with the tuning law given by (20) when the learning rate and γ are all satisfied this following requirement

$$0 < \eta_\beta < \frac{2}{\gamma + \Delta u^2(k-1)}. \tag{21}$$

Proof: Rearrange (20) in vector format, we obtain

$$\beta(k+1) = \left[1 - \eta_\beta[\gamma + \Delta u^2(k-1)]||F(k)||^2(k)\right]\beta(k) + \xi(k), \tag{22}$$

when $\xi(k)$ stands for the remain terms which are not related on $\beta(k)$. Let $A = 1 - \eta_\beta[\gamma + \Delta u^2(k-1)]||F(k)||^2(k)$, we have

$$\beta(k+1) = A\beta(k) + \xi(k). \tag{23}$$

$||F(k)||^2(k)$ is the multiplication of two membership grades, thus $0 < ||F(k)||^2(k) \leq 1$ and substitute with (21) it can be obtained

$$A < 1. \tag{24}$$

□

The relation of the learning rate η_{beta} and γ obtained by this lemma can be used to support the design of both parameters which will be demonstrated in the next section. Furthermore, this result can be valid even $\Delta u(k-1) = 0$ or the previous control effort is constant.

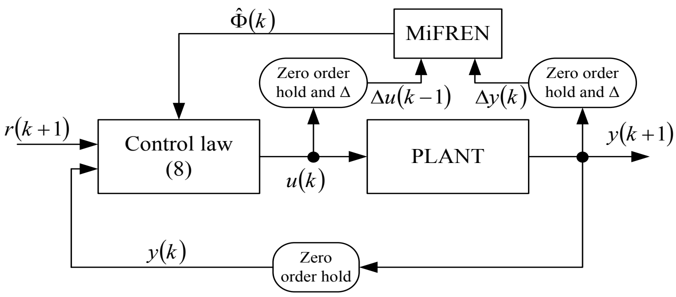

Figure 2: Control system configuration.

Furthermore, the block diagram illustrated in Fig. 2 represents the design concept and signal flow.

5. Design example and simulation results

The nonlinear discrete-time system which is selected to demonstrate the performance is described as

$$y(k+1) = \sin(y(k)) + u(k)[5 + \cos(y(k)u(k))], \tag{25}$$

when $y(k)$ denotes the output and $u(k)$ stands for the control effort. The desired trajectory $r(k)$ is given by this following:

$$r(k) = 2\sin(\frac{2\pi k}{100}). \tag{26}$$

To design the controller which is suitable for the system described in (25), in this case, we have the range of Δy and Δu in ± 2, thus the membership functions can be defined by Fig. 3 and 4 for Δy and Δu, respectively.

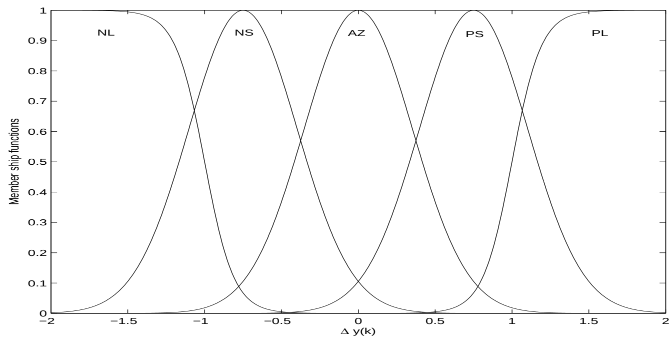

Figure 3: Membership function of $\Delta y(k)$.

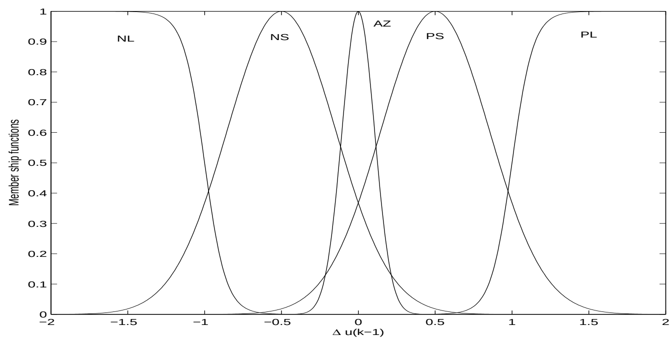

Figure 4: Membership function of $\Delta u(k-1)$.

According to (9) and (21), those designed parameters can be given as $\rho = 0.35$, $\lambda = 0.75$, $\eta_\beta = 1.5$ and $\gamma = 0.1$.

The adjustable parameters β_i can be defined by Table 2 regarding to the IF-THEN rules.

The tracking performance can be demonstrated in Fig. 5 as plots of the desired trajectory $r(k)$ and the system output $y(k)$. Finally, in Fig.6, the control effort is shown.

The demonstration of robustness is introduced next. In this setup, the time varying disturbance $d_1(k)$ is added in the original system (25) as

$$y(k+1) = \sin(y(k)) + u(k)[5 + d_1(k) + \cos(y(k)u(k))]. \tag{27}$$

such as constant parameters and $\beta_i(1)$ are same as the previous test. Fig. 7 represents the tracking performance and time varying $d_1(k)$. The control effort $u(k)$ is illustrated in Fig. 8.

For more complicity, another disturbance $d_2(k)$ is also included from the nominal system (25) as

$$\begin{aligned} y(k+1) &= \sin(y(k) + d_2(k)) \\ &\quad + u(k)[5 + d_1(k) + \cos(y(k)u(k))]. \end{aligned} \tag{28}$$

With all same initial settings, the tracking performance and disturbances can be shown in Fig. 9 according the control effort displayed in Fig. 10.

$\Delta y(k)$					Δu
PL	PS	AZ	NS	NL	$(k-1)$
$\beta_1(1)$ $=3$	$\beta_6(1)$ $=1.5$	$\beta_{11}(1)$ $=0.2$	$\beta_{16}(1)$ $=0$	$\beta_{21}(1)$ $=0$	PL
$\beta_2(1)$ $=4$	$\beta_7(1)$ $=2.5$	$\beta_{12}(1)$ $=0.5$	$\beta_{17}(1)$ $=0$	$\beta_{22}(1)$ $=0$	PS
$\beta_3(1)$ $=0.1$	$\beta_8(1)$ $=0.1$	$\beta_{13}(1)$ $=0.1$	$\beta_{18}(1)$ $=0.1$	$\beta_{23}(1)$ $=0.1$	AZ
$\beta_4(1)$ $=0$	$\beta_9(1)$ $=0$	$\beta_{14}(1)$ $=0.5$	$\beta_{19}(1)$ $=2.5$	$\beta_{24}(1)$ $=4$	NS
$\beta_5(1)$ $=0$	$\beta_{10}(1)$ $=0$	$\beta_{15}(1)$ $=0.2$	$\beta_{20}(1)$ $=1.5$	$\beta_{25}(1)$ $=3$	NL

Table 2: IF-THEN rules with initial parameters $\beta_i(1)$

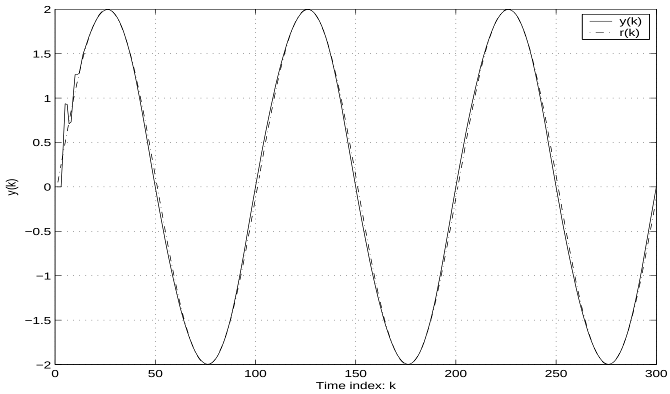

Figure 5: Tracking performance $y(k)$ and $r(k)$.

Remark: Any information related on those disturbances is not necessary to incorporate to the controller. That means the original controller designed based on the knowledge about the nominal plant is enough to handle those disturbance.

6. Conclusion

The model-free adaptive controller with the estimation of pseudo-partial derivation is proposed in this article. The estimation of PPD is implemented by a self-adjustable network called MiFREN. The initial setting of MiFREN's structure can be given by the human knowledge according to the controlled plant and the relation of plant's input-output within IF-THEN rules format. Moreover, all adjustable parameters inside MiFREN are begun by the knowledge of the plant directly. That can improve the system performance at the beginning. Other fixed parameters are designed by proved lemmas to guarantee the convergence. The simulation system demonstrates the design example and the effectiveness of the proposed algorithm. Both nominal system and disturbed plant have been considered to validate the controller performance and robustness.

Acknowledgment

The author would like to thank CONACyT (Project # 84791) and CINVESTAV-IPN for the financial support through this work.

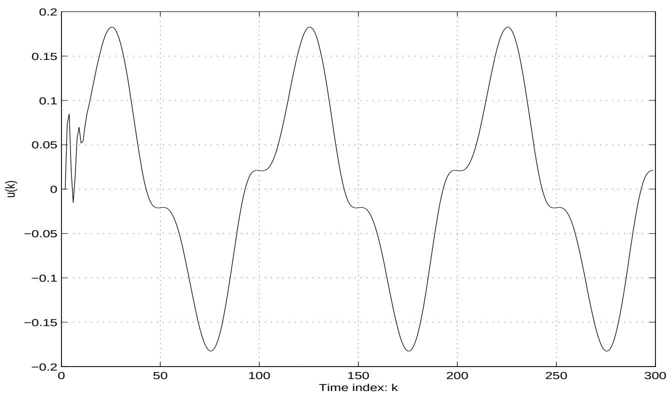

Figure 6: Control effort $u(k)$.

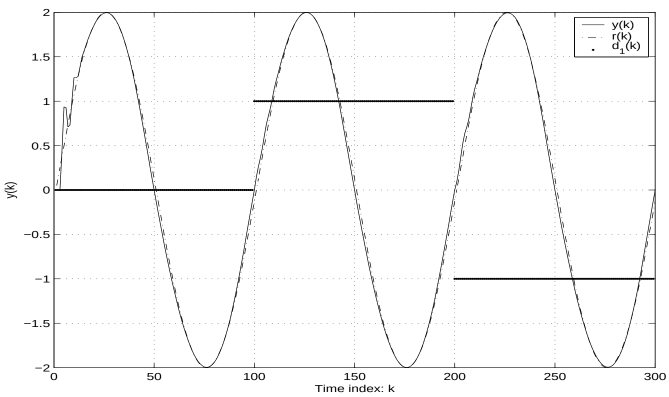

Figure 7: Tracking performance $y(k)$ and $r(k)$ with disturbance $d_1(k)$.

References

[1] B. Zhang and W. D. Zhang, "Adaptive predictive functional control of a class of nonlinear systems," *ISA Trans.*, vol. 45, no. 2, pp. 175-183, 2006.

[2] D. Meng, Y. Jia, J. Du and F. Yu, "Data-driven control for relative degree systems via iterative learning," *IEEE Trans. Neural Networks.*, vol. 22, no. 12, pp. 2213-2225, Dec. 2011.

[3] Z. S. Hou and S. T. Jin, "A novel data-driven control approach for a class of discrete-time nonlinear systems," *IEEE Trans. Control Syst. Technol.*, vol. 19, no. 6, pp. 1549-1558, Nov. 2011.

[4] L. D. S. Coelho, W. P. Marcelo, R. S. Rodrigo, and A. A. R. Coelho, "Model-free adaptive control design using evolutionary-neural compensator," *Expert Syst. Appl.*, vol. 37, no. 1, pp. 499-508, 2010.

[5] B. Hahn and K. R. Oldham, "A model-free ONÚOFF iterative adaptive controller based on stochastic approximation," *IEEE Trans. Control Syst. Technol.*, vol. 20, no. 1, pp. 196-204, Jan. 2012.

[6] C. Treesatayapun and S. Uatrongjit, "Adaptive controller with Fuzzy rules emulated structure and its applications'," *Engineering Applications of Artificial Intelligence*, Elsevier, Volume 18, P. 603-615, 2005

[7] Z. S. Hou and S. T. Jin, "Data-driven model-free adaptive control for a class of MIMO nonlinear discrete-time systems," *IEEE Trans. Neural Networks.*, vol. 22, no. 12, pp. 2173-2188, Dec. 2011.

[8] C. Treesatayapun, "A discrete-time stable controller for an omnidirectional mobile robot based on an approximated model," *Control Engineering Practice*, Elsevier, vol. 19, pp. 194-230, 2011

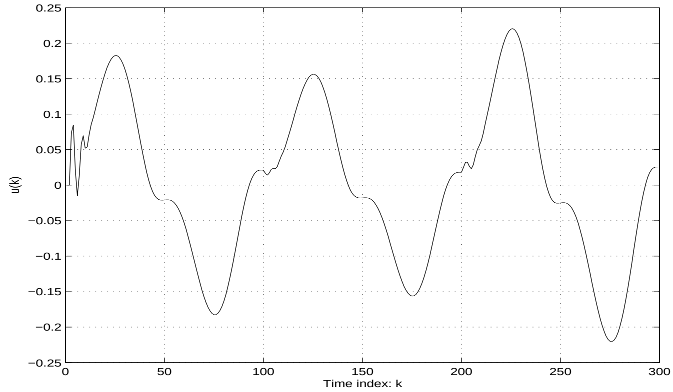

Figure 8: Control effort $u(k)$ with disturbance $d_1(k)$.

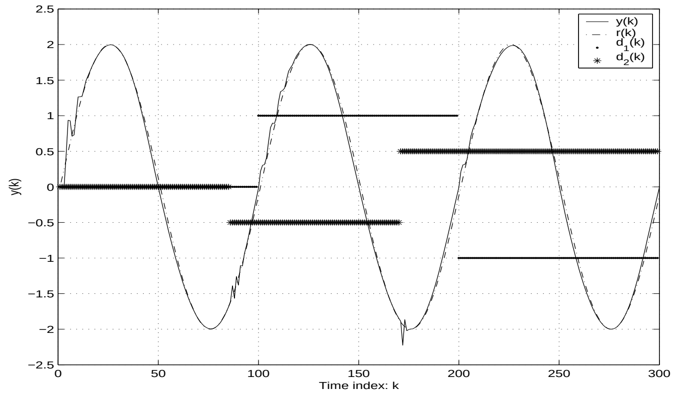

Figure 9: Tracking performance $y(k)$ and $r(k)$ with disturbances $d_1(k)$ and $d_2(k)$.

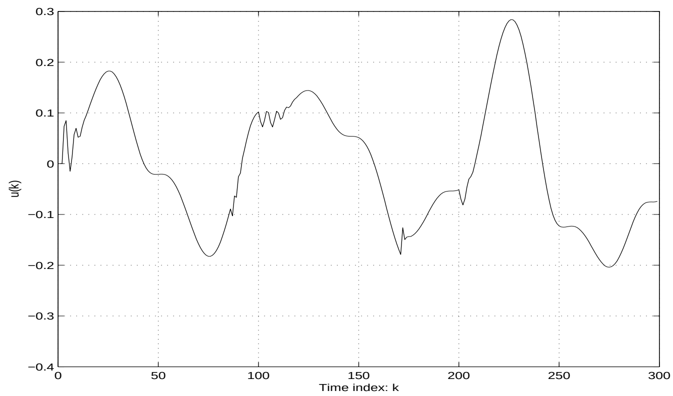

Figure 10: Control effort $u(k)$ with disturbances $d_1(k)$ and $d_2(k)$.