2005 American Control Conference
June 8-10, 2005. Portland, OR, USA

Self-Organizing Intelligent System and Its Application for Hard Disk Drive Control

Chunshien Li, Kuo-Hsiang Cheng, Jin-Long Chen, Chih-Ming Chen

***Abstract*—An intelligent system with self-organizing ability is discussed in this paper. The intelligent system is based on computational models of fuzzy inference, neural-processing, and machine learning. The concept of pseudo errors is utilized in the study. Clusters are generated from the pseudo-error information, and they are corresponding to fuzzy rules of the self-organizing neuro-fuzzy system (SO-NFS). In this paper, a learning scheme is proposed for hard disk drive (HDD) motion control. The SO-NFS is used as a controller to the HDD servo system for both the track-seeking and the track-following motion control. A reference speed-displacement curve is used to guide the control process. Good performance of the proposed approach is observed.**

I. INTRODUCTION

IN resent years, intelligent systems and applications have become an interesting issue and have attracted a growing attention of researchers and engineers [3-6]. For intelligent system, the problems of system organization and learning need to be intensively discussed. In the published works, several schemes have been proposed for system self-evolution [3][6]. Self-evolving systems own the learning and adapting abilities to specific application, and can alleviate interference of human-operation and human-design. For control application, a self-organizing intelligent system can be used to capture the essentials of control strategy and to optimize the system performance. With the human-like fuzzy reasoning, fuzzy logic can be used to transform the engineering experience and the knowledge about system dynamics into linguistic description [2][9]. To obtain suitable system structure and parameters, the system is expected to evolve itself. For system optimization, learning algorithms such as genetic algorithm (GA) [7] and back propagation (BP) [2] are the most popular. However, there are some inevitable drawbacks in these algorithms. The evolution-based GA algorithm needs a considerable computation resource because of the mass operation of chromosomes. The gradient-based BP algorithm back propagates the derivative information of the controlled system to update controller parameters. But, the derivative information may not be available in real application.

An intelligent control system with self-organizing ability is discussed in this paper. In the process of rules extraction, the concept of pseudo-errors, providing the priori information of the controlled system for the approach, is adopted in the construction of the knowledge base of the intelligent neuro-fuzzy system. Clusters are generated from the pseudo-error information, and they are corresponding to fuzzy rules of the self-organizing neuro-fuzzy system (SO-NFS). The well-known random optimization (RO) [6] algorithm is utilized to evolve the optimal set. A hard disk drive (HDD) motion control application is used for verifying the feasibility of the proposed approach.

II. SELF-ORGANIZING INTELLIGENT SYSTEM

Consider an M-input fuzzy inference system which comprises K IF-THEN rules, the linguistic description can be given as:

IF (x_1 is s_1^i ($h_1(t)$)) and…and (x_M is s_M^i ($h_M(t)$))

THEN $z^i(t) = a_0^i + a_1^i h_1(t) + ...a_M^i h_M(t)$ (1)

for i= 1, 2, …, k and j= 1, 2, …, M, where the x_k is the k-th input variables, s_j^i the i-th fuzzy set for the j-th input dimension, $h_j(t)$ the j-th crisp input at time t, and $z^i(t)$ the

Chunshien Li is with the Department of Electrical Engineering, Chang Gung University, Tao-Yuan, 333 Taiwan, R.O.C., (Telephone No.: +886-3-2118800; Fax:+886-3-2118026; e-mail: jamesli@mail.cgu.edu.tw).

Kuo-Hsiang Cheng is with the Department of Electrical Engineering, Chang Gung University, Tao-Yuan, 333 Taiwan, R.O.C., (e-mail: D9021010@stmail.cgu.edu.tw).

Jin-Long Chen is with the Department of Electrical Engineering, Chang Gung University, Tao-Yuan, 333 Taiwan, R.O.C., (e-mail: D9121008@stmail.cgu.edu.tw).

Chih-Ming Chen is with the Department of Electrical Engineering, Northern Taiwan Institute of Science and Technology, No. 151, Yide st.,Beitou, Taipei, 112 Taiwan, R.O.C., (e-mail:jmchen@ntist.edu.tw).

This work is supported by the National Science Council, Taiwan, Republic of China, under Grant NSC93-2213-E-182-023.

i-th rule output at time t. In this paper, a layered feedforward neural network [6] is used to represent the SO-NFS as shown in Fig. 1. In $L2$, the fuzzy sets given by Gaussian functions and the layer constitution in $L3$ is partially connected. In the SO-NFS, K fuzzy sets are in each input dimension.

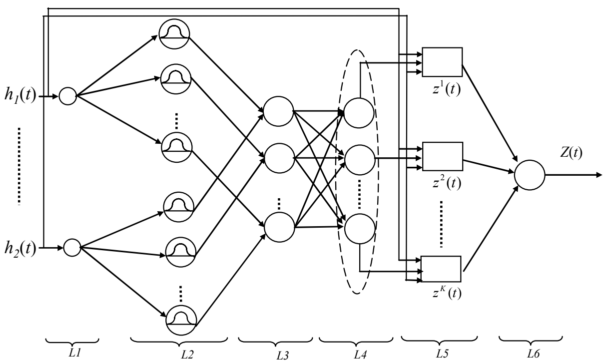


Fig. 1. Self-organizing neuro-fuzzy system.

To self-explore the controlled system behavior, the concept of pseudo errors [6] is adopted. Pseudo-targets are designed and viewed as the pseudo control desired targets of application, which can provide the priori-information for a plant. With appropriately given pseudo targets, the pseudo errors information can be generated sufficiently enough in the application domain of interest. Once pseudo-errors data are sufficiently generated, the potential behavior of the plant is sufficiently revealed. Based on the observed pseudo-errors information, the structure identification can proceed to capture the potential fuzzy rules for the plant. The generation process of pseudo errors is illustrated in Fig. 2.

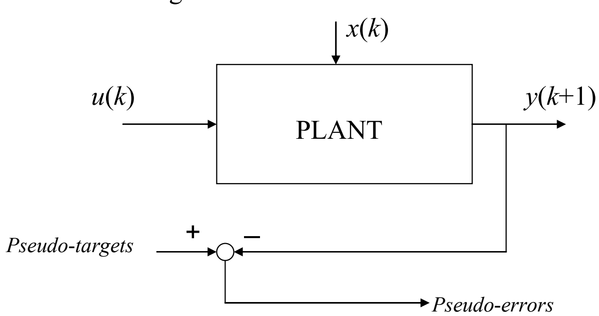


Fig. 2. Generation process of pseudo-errors.

To self-construct the SO-NFS, a statistical regression model with clustering organization is used to integrate the pseudo-error information. With the regression model, the distribution trend of the observed data can be expressed as follows.

$$y_m = \alpha_0 + \alpha_1 x_m + \alpha_2 {x_m}^2 + \alpha_3 {x_m}^3 + \varepsilon_m \tag{2}$$

where (x_m, y_m) is the observed data, ε_m the model error. The solution to the above equation results in the least square estimation of the α_0, α_1, α_2, and α_3. With the estimation function, the clusters can be generated on the estimated curve. Each cluster can be regarded as an antecedent fuzzy rule, which comprises membership functions of fuzzy sets of input dimensions. A generated cluster for the SO-NFS can be expressed as follows.

$$C^i(H(t)) = \wedge\{[\exp(-(\frac{h_j(t) - m_j^i}{\varpi_j^i})^2)],\ j = 1,2,...,M\} \tag{3}$$

for i=1,2,…,K, where $\wedge$ denotes the operation of "fuzzy-and", which can be calculated using any t-norm operator; superscript i indicates the i-th cluster; and $h_j(t)$ is the j-th element in the input vector $H(t)$, serving as a crisp input in the j-th dimension of a cluster. The value of $C^i(H(t))$ is interpreted as the firing strength of the i-th cluster for the $H(t)$ at time t. In the SO-NFS, Gaussian function is used to describe fuzzy set, and m_j^i and ϖ_j^i are the mean and spread in the j-th dimension. For example, a two-dimensional cluster is illustrated in Fig. 3.

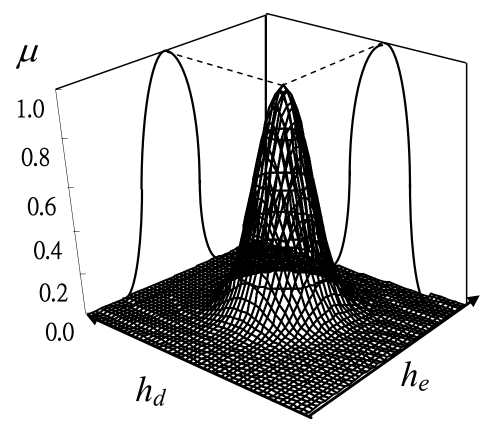


Fig. 3. A two-dimensional SO-NFS cluster constitution.

Since the Takagi-Sugeno fuzzy model [9] is used in the control system, only the input space partition is concerned. The coefficients of all consequents in the rule base can be collected to form a matrix $\boldsymbol{A}$, given as

$$\boldsymbol{A} = \begin{bmatrix} a_0^1 & a_1^1 & \cdots & a_M^1 \\ a_0^2 & a_1^2 & \cdots & a_M^2 \\ \vdots & \vdots & \ddots & \vdots \\ a_0^K & a_1^K & \cdots & a_M^K \end{bmatrix} \tag{4}$$

The parameters of the spreads and means of the antecedents in the rule base can be collected together as follows.

$$\Sigma = \{\Sigma^i, i = 1,2,...,K\},\ \Delta = \{\Delta^i, i = 1,2,...,K\} \tag{5}$$

$$\Sigma^i = [m_1^i, m_2^i, ..., m_M^i], \Delta^i = [\varpi_1^i, \varpi_2^i, ..., \varpi_M^i] \quad (6)$$

The parameter set of the SO-NFS can be expressed as follows.

$$\boldsymbol{W} = (\Sigma, \Delta, \boldsymbol{A}) \quad (7)$$

The SO-NFS can be applied to control application. The cost function used in the paper is defined as follows.

$$f(W) = \sum_{i=1}^{N_{pattern}} \left\{ factor1 \sum_{k=1}^{p} (r_i(k) - y(k))^2 + factor2 \sum_{k=p+1}^{n} (r_i(k) - y(k))^2 \right\} \quad (8)$$

where the $N_{pattern}$ is the number of training patterns, r_i the desired output as the i-th training pattern, y the plant output, *factor* 1 and *factor* 2 the weighting factors, n the number of time steps involved in the cost function, and p the time to switch the weighting factor from *factor* 1 to *factor* 2. The weighted cost function can make the SO-NFS controller to trace the target more precisely when the plant output gets close to the target. For the simplicity of understanding, the set of parameters for the controller is viewed as a point in the parameter space, and an appropriate set of parameters can be learned after the process of parameters identification. The derivative-free RO algorithm is used to search for the interpolation region of the parameter space [6] to increase its searching performance. Moreover, a jumping mechanism is designed in the RO algorithm to prevent the searching process from being trapped at a local minimum. The interpolation points provide a search around the current point for fine turning and the jump mechanism takes a wider region under consideration. In the point of convergence, RO algorithm has been proved to convergent to the point of global minimum with the probability of 1 in a compact parameter space. The Gaussian probability distribution is used for random number generation, and a normal-distributed random number is generated to adjust the system parameters gradually. The RO is also powerful for its simplicity, convenience and derivative-free property. The RO learning approach is flexible for the learning application. This is a significant merit, especially when the engineers encountered complex problems. In general, the RO method used in the paper is expected to search and move to a better point of the parameter space with a smaller cost value, and around the current point the exploration continues. If the RO fails to search a better point, the interpolation points are examined for better candidate point. If the RO algorithm is trapped at a local minimum for a period of time, the "jump" mechanism restarts the search by jumping randomly to a new point and the searching continues.

III. Hard Disk Drive System

The storage technologies of accessing efficiency and positioning accuracy have become a challenge to engineers [1][8][10]. The structure of HDD is given in Fig. 4.

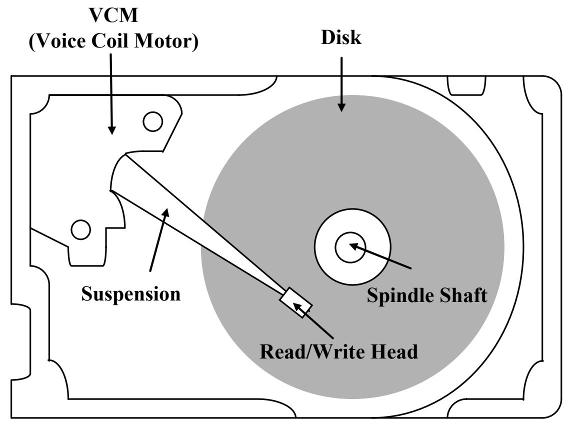


Fig. 4. HDD model.

In this paper, a learning scheme is proposed for HDD motion control. The control system is trained to trace the acceleration patterns first, and the acceleration and deceleration patterns are given in the second phase of training. With the advantage of fuzzy inference theory, the generalization of the control policy can be achieved in training processes. In order to respond rapidly to the requirements, the R/W head of a HDD is demanded to go all out from the start track to the target track in the track seeking process, and the head should be settled down at the target track with the smoothness in the mode switching. A design of reference curve [5] for the R/W head motion for the track seeking/following control is proposed in this approach. The corresponding speed-trajectory reference curve is designed, varying with the distance for the R/W head to move. The maximum acceleration and deceleration are given with $8.40 \times 10^4\ cm/s^2$. The reference curve plays a vital role to describe the relationship between the speed and the displacement of the R/W head. With the Newton's motion law, the total distance S_T can be given as follows.

$$S_T = S_{acc} + S_{\max-speed} + S_{dec} + S_{creep} \quad (9)$$

where S_{acc}, $S_{max\text{-}speed}$, S_{dec}, and S_{creep} are the distance of the R/W head for acceleration, maximum speed motion, deceleration, and linear creep to the target track, respectively. The reference signal for the speed of the R/W head is given using the distance-speed motion reference curve, which is designed for radial motion of the R/W head, according to the specifications of the servo system, such as maximum head speed, acceleration, and deceleration. The maximum head speed V_{max} is decided according to the distance to move if not excessing the maximum speed given in the specifications. Otherwise, the head is kept with the speed value in the specifications. The lengths are varying with the total distance S_T between the start track and the target track. In the approach, V_{creep} and S_{creep} for R/W head creeping to the target track are given and fixed. For the specifications of the reference curve, S_{min}, V_{max}^{spec}, V_{creep}, S_{screep} [5], are given with 200 μm, 240 cm/s, 5 cm/s, and 20 μm, respectively. The reference curve design is given in [5]. The profile of the linear creep curve is shown in Fig. 5.

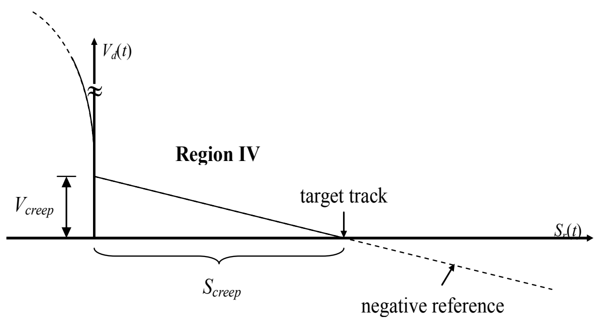


Fig. 5. Profile of the region IV of the reference curve.

With the reference curve, the desired speed for the R/W head, $V_d(t)$, is obtained for the current displacement of movement. Based on the HDD model given in [8], the information of position and speed of the R/W head in radial motion is transformed from the angular motion with $\theta(t)$ and angular speed $\dot{\theta}(t)$. The transfer functions are given as follows.

$$D_r(\theta)=\sqrt{L_d^{\ 2}+L_{arm}^{\ 2}-2L_d L_{arm}\cos(\theta)} \quad (10)$$

$$S_r(t)=\left\| D_r(\theta(t))-D_r(\theta_S) \right\| \quad (11)$$

$$V_r(t)=direction\times\frac{L_d L_{arm}\sin\theta(t)}{\sqrt{L_d^2+L_{arm}^2-2L_d L_{arm}\cos\theta(t)}}\times\dot{\theta}(t) \quad (12)$$

where θ_S is the angular position of the start track, $S_r(t)$ the displacement from the spindle shaft of the disk at time t, $V_r(t)$ the speed of the R/W head in radial motion, and $direction=\begin{cases}+1, & if\ \theta_T>\theta_S\\ -1, & otherwise\end{cases}$, θ_T is the angular position of the target track. Note that θ is positive if clockwise. The sizes of L_d=6 cm and L_{arm}=6.1 cm are used in the paper.

IV. Implementation of the SO-NFS to the HDD System

The SO-NFS is utilized as a controller to the HDD system for both the track-seeking and the track-following motion control. The closed-loop control scheme is shown in Fig. 6. Two input linguistic variables are selected for the SO-NFS [6], and they are error and its derivative, denoted as *err* and *derr*, whose base variables are denoted as h_e and h_d, respectively.

$$h_e(t)=V_d(t)-V_r(t) \text{ and } h_d(t)=h_e(t)-h_e(t-1) \quad (13)$$

The control desired speed $V_d(t)$ is obtained from the reference curve. The parameter λ of 0.4 is used to scale down the output of the SO-NFS and a saturation function is used to restrict the amount of input to the HDD system to comply with the input limitation between $\pm$9 V.

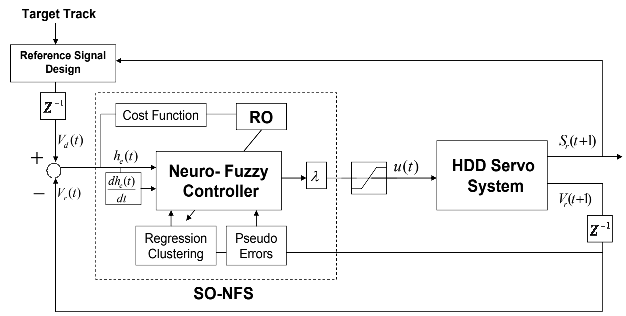


Fig. 6. Closed-loop control using SO-NFS.

To explore pseudo errors information of the hard disk drive system, appropriate pseudo-target patterns are used, according to the motion specification of the hard disk drive. A few of speed patterns are designed so that the exploration of the behavior of hard disk drive system can be done as sufficiently as possible. The HDD is given a random input at each period of sampling time, and the output is compared with these pseudo-targets to generate pseudo errors. The random input is limited in between $\pm$5 V. The speed pseudo target patterns for generation of pseudo-errors are given as follows.

Speed pseudo target patterns: [0, 5, 10, 15] (*cm/s*)

The generated pseudo-errors is shown in Fig. 7. There are 9 clusters generated for the SO-NFS. The Gaussian membership functions are defined simultaneously using the centers and sizes of the clusters.

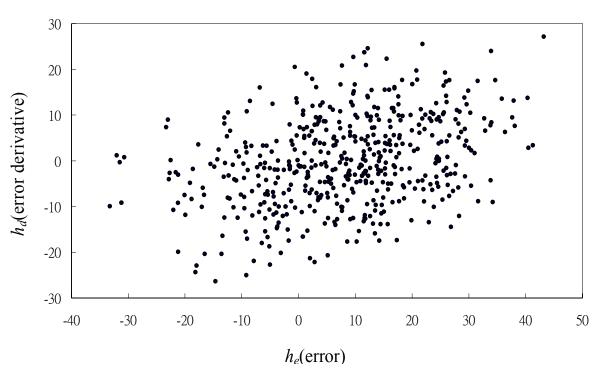


Fig. 7. Generated pseudo-errors of HDD.

The parameter learning process is put into two learning sessions. In the real situation the R/W head is accelerated first, and is decelerated to the creep speed. The driving process reveals that the controller should be trained for acceleration first, and then for deceleration. In the two-session learning, two different cost functions are designed and two different sets of pseudo targets are chosen for the learning purpose.

(a) first-session learning for acceleration

At the beginning, the SO-NFS is trained for accelerating the R/W head to trace the maximum speed. The training patterns are given as follows.

***Speed patterns*:** [0.001, 0.01, 0.1, 5, 10, 20, 30] (*cm/s*)
Each training pattern is presented as the desired speed for the HDD acceleration, and parameter is evolved to accelerate the R/W head from zero speed. There are seven patterns given to the controller and the speed error information is accumulated to form the cost function. For the consideration of control accuracy, the small speed patterns such as 0.001 and 0.01, 0.1 (cm/s) are chosen. The cost function in session 1 is given as follows.

$$^{1}f(W)=\sum_{i=1}^{pattern1}\left\{factor1\sum_{k=1}^{ftime}(r_i(k)-y_i(k))^2+factor2\sum_{k=ftime+1}^{n}(r_i(k)-y_i(k))^2\right\} \tag{14}$$

where W is the parameter set of the SO-NFS, $pattern1$ is the number of patterns for acceleration and is set to 7; $factor1$=1, $factor2$=100, $ftime$=100, and n=150.

(b) second-session parameter learning

The complete procedure of HDD head motion control is implemented for the second-phase parameter learning for acceleration and deceleration. The training patterns in session 2 are given as follows.

***Speed patterns*:**
r^{acc}: [0.01, 0.1, 5, 10, 20, 30] (*cm/s*) for acceleration,
r^{dec}: [15, 10, 5, 0.01, -0.01, 0] (*cm/s*) for deceleration.
The cost function is given as follows.

$$\begin{aligned}^{2}f(W)=\sum_{i=1}^{pattern2}\{&factor3\sum_{k=1}^{ftime}(r_i^{acc}(k)-y_i(k))^2+factor4\sum_{k=ftime+1}^{n}(r_i^{acc}(k)-y_i(k))^2)+\\&factor3\sum_{k=n+1}^{n+ftime}(r_i^{dec}(k)-y_i(k))^2+factor4\sum_{k=n+ftime+1}^{2n}(r_i^{dec}(k)-y_i(k))^2\}\end{aligned} \tag{15}$$

where $pattern2$ is the number of patterns for acceleration and deceleration and is set to 6, n=150, $ftime$=100, $factor3$ and $factor4$ are defined as weighting factors and given as 1 and 100, respectively. In this stage of learning, the acceleration and the deceleration are learned simultaneously. With the RO algorithm [6], the parameter set of the SO-NFS is evolved to minimize the cost function. In parameter learning process, there are 1000 learning cycles for session 1 and 1500 learning cycles for session 2. In this study, the performance of SO-NFS is compared with the conventional PD and PID approaches, where the controller parameters are given as K_p =80, K_d =16 for PD controller and K_p =80, K_I =32, K_d =4 for PID controller by empirical design. For the positioning performance from 1 to 7000 tracks, the SO-NFS approach is superior to two compared methods. The performance comparison is given in Fig. 8. The results for 300 tracks (1183.5 μm) are shown in Fig. 9. Superior performance by the proposed SO-NFS in positioning accuracy and motion efficiency is observed.

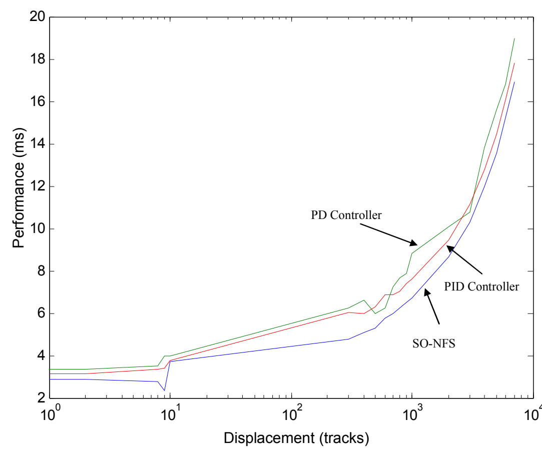


Fig. 8. Performance comparison of SO-NFS with PD and PID.

V. Discussion and Conclusion

The two-session learning process is given. The training patterns served as the possible desired speed targets in the first-session learning process for acceleration from zero speed. By the training, the controller possesses the ability to accelerate, while the capability of deceleration needs to be further learned in the second learning session. With the intuitive and straightforward search scheme, the random optimization (RO) algorithm, which is a derivative-free approach, is used in system learning process. It provides an efficient and convenient method to overcome the drawbacks of the gradient-based algorithms. In the system structure identification process, a basic concept of pseudo errors for the SO-NFS to self-explore the dynamic behavior of the HDD is used. With the cluster-oriented rule extraction, the potential information of the system dynamics is analyzed to form the initial control policy. The generated clusters are viewed as multi-dimensional fuzzy membership functions. Each cluster represents logical combination of control reasoning in the semantic inference. The RO algorithm holds the merits for parameter learning of both coarse and fine search, with the mechanism of interpolation search and jump. The reference motion curve is designed for the HDD servo system. The proposed neuro-fuzzy soft-computing approach not only simplifies the complexity of control process, by which the same SO-NFS controller is utilized for both track seeking mode and track following mode of the HDD servo system, but also improves the accessing efficiency in terms of accessing time and positioning accuracy of the R/W head. The SO-NFS approach is superior to the results in [8] about 2 to 16% improved in accessing time.

REFERENCES

[1] G. Guo, R. Chen, T.S. Low and Y. Wang, "Optimal control design for hard disk drive servo systems," *IEE Proc.-Control Theory Appl*., vol. 149, no. 3, pp. 237-242, May 2002.

[2] S. Horikawa, T. Furuhashi, Y. Uchikawa, "On Fuzzy Modeling Using Fuzzy Neural Networks with the Back-Propagation Algorithm." *IEEE Transactions on Neural Networks*, vol. 3, no. 5, pp.801-806, September 1992.

[3] C. F. Juang and C. T. Lin, "An on-line self-constructing neural fuzzy inference network and its applications," *IEEE Transactions on Fuzzy Systems*, vol. 6, no. 1, pp. 12-31, February 1998.

[4] C. T. Lin, "A neural fuzzy control system with structure and parameter learning," *Fuzzy Sets and Systems*, vol. 70, no. 2-3, pp. 183-212, 1995.

[5] C. Li and C.Y. Lee, "Fuzzy motion control of an auto-warehousing crane system," *IEEE Transactions on Industrial Electronics*, vol. 48, no. 5, pp. 983-944, October 2001.

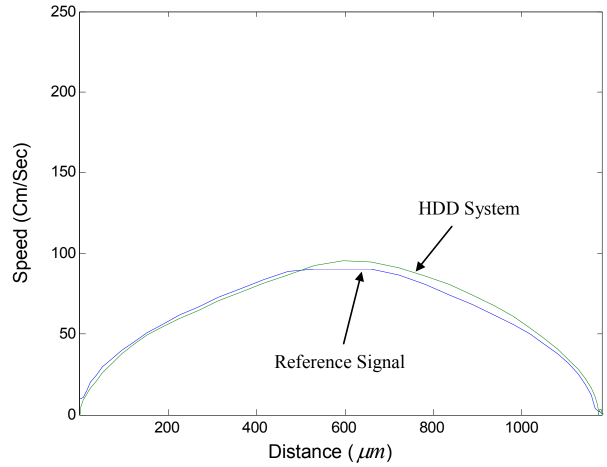
[6] C. Li and C. Y. Lee, "Self-organizing neuro-fuzzy system for control of unknown plants," *IEEE Transactions on Fuzzy Systems*, vol. 11, no. 1, pp. 135-150, Feb. 2003.

[7] F. H. F. Leung, H. K. Lam, S. H. Ling, P. K. S. Tam, "Tuning of the Structure and Parameters of a Neural Network Using an Improved Genetic Algorithm." *IEEE Transactions on Neural Networks*, vol. 14, no. 1, pp. 79-88, January 2003.

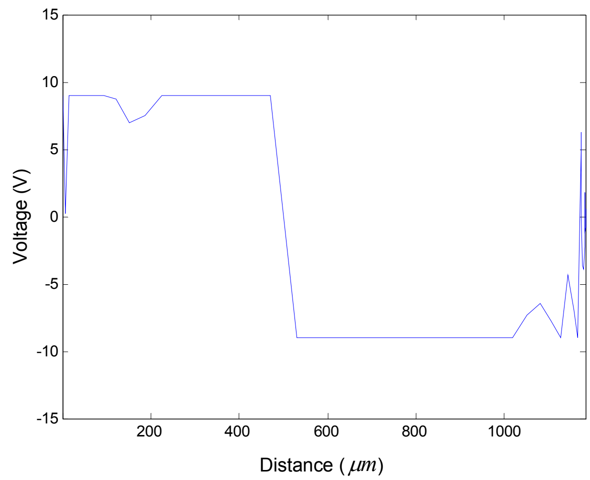
[8] H. Susumu, H. Takeyori, Yi Li and T. Masayoshi, "Novel reference signal generation for two-degree-of –freedom controllers for hard disk drives," *IEEE Transactions on Mechatronics*, vol. 5, no. 1, pp. 73-78, March 2000.

[9] Takagi and M. Sugeno, "Fuzzy identification of systems and its applications to modeling and control," *IEEE Transactions on Systems, Man and Cybernetics*, SMC-15(1), pp. 116-132, January/February 1985.

[10] D. Q. Zhang and G. X. Guo, "Discrete-time sliding mode proximate time optimal seek control of hard disk drives," *IEE Proc.- Control Theory Appl.*, vol. 147, no. 4, pp. 440-446, July 2000



(a)



(b)

Fig. 9. 300 tracks (1183.5 μm) seeking/following control of the R/W head of HDD. (a) speed-distance curve. (b) input voltage-distance curve.