C. Case C: L-Order Filter with Arbitrary Intermediate Power Levels

Consider now Λ has diagonal elements which can take on arbitrary power levels such that $\{\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_{L-1} \geq \lambda_L\}$. It follows from (13) that

$$\frac{\overline{\lambda}_i}{\lambda_i} = \int_0^\infty \frac{d\beta}{(1+2\beta\lambda_i)\sqrt{|(\boldsymbol{I}+2\beta\Lambda)|}} \tag{28}$$

where $|\cdot|$ is determinant operator. It can be easily shown that $\overline{\lambda}_1 \geq \overline{\lambda}_2 \geq \cdots \geq \overline{\lambda}_L$, and the condition number of $\overline{\Lambda}$ is always no greater than that of Λ

$$\frac{\overline{\lambda}_1}{\lambda_1} \leq \frac{\overline{\lambda}_L}{\lambda_L} \Rightarrow \frac{\overline{\lambda}_1}{\overline{\lambda}_L} \leq \frac{\lambda_1}{\lambda_L} \Rightarrow C(\overline{\Lambda}) \leq C(\Lambda). \tag{29}$$

In addition to this bound, a conjecture is also given which sets bounds on the condition number of $\overline{\Lambda}$. By fixing only the maximum and minimum unnormalized eigenvalues while allowing the intermediate eigenvalues to take on arbitrary values, this conjecture states that

$$\begin{aligned} &C(\overline{\Lambda})_{\{\lambda_1=\lambda_2\cdots=\lambda_{L-1}\leq\lambda_L\}} \\ &\quad\leq C(\overline{\Lambda})_{\{\lambda_1\geq\lambda_2\cdots\geq\lambda_{L-1}\geq\lambda_L\}} \\ &\quad\leq C(\overline{\Lambda})_{\{\lambda_1=\lambda_2\cdots=\lambda_{L-1}\geq\lambda_L} \end{aligned} \tag{30}$$

where the upper and lower bounds are the condition numbers defined in the previous case. This conjecture has also been verified with extensive computer simulations, which leads to an important observation: given the same condition number of the LMS adaptation, the NLMS algorithm can potentially converge much faster when the majority of intermediate eigenvalues contain low power values. It should be remembered that these interpretations are based on *zero-mean* Gaussian distribution of input samples.

III. Remarks

This paper has presented new interpretation on the relative convergence rate performance of the NLMS adaptation to that of the LMS algorithm, under the assumption that the input samples are zero-mean Gaussian distributed. To apply these results for analyzing existing neural models, the effect of sample bias remains an important subject to be studied although simulation results suggest that the improvement in convergence rate reduces as the bias increases and the normalization term has less mean-squared fluctuation. The difference in the convergence rate between the LMS and NLMS algorithms is more pronounced for a smaller filter length and larger LMS condition number. This paper has not studied the steady state convergence performance both in mean and mean-squared cases although these important aspects have been considered rigorously in [2] and [9]. However, in their work, the excessive mean-squared output error was evaluated with respect to the unnormalized Wiener solution, rather than the normalized one given in (8). It seems intuitive that the misadjustment based on the minimum mean-squared solution would yield a larger value as compared to that in the LMS case, without regard to the fairness of such comparison.

References

[1] P. E. An, M. Brown, and C. J. Harris, "A global gradient noise covariance expression for stationary real Gaussian inputs," *IEEE Trans. Neural Networks*, vol. 6, pp. 1549–1551, Nov. 1995.

[2] N. Bershad, "Analysis of the normalized LMS algorithm with Gaussian inputs," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-34, pp. 793–806, 1986.

[3] M. Brown and C. J. Harris, *NeuroFuzzy Adaptive Modeling and Control, Hemel Hempstead*. Englewood Cliffs, NJ: Prentice-Hall, 1994.

[4] S. Haykin, *Adaptive Filter Theory*, 2nd ed. Englewood Cliffs, NJ: Prentice-Hall, 1991.

[5] S. Kaczmarz, "Angenaherte Auflösung von Systemen Linearer Gleichugen," *Bull. Int. Acad. Pol. Sci. Lett.*, pp. 355–357, 1937 (in German).

[6] J. I. Nagumo and A. Noda, "A learning method for system identification," *IEEE Trans. Automat. Contr.*, vol. AC-12, pp. 282–287, 1967.

[7] D. T. M. Slock, "On the convergence behavior of the LMS and the normalized LMS algorithms," *IEEE Trans. Signal Processing*, vol. 41, pp. 2811–2825, 1993.

[8] M. R. Spiegel, *Mathematical Handbook*, Schaum's Outline Series. New York: McGraw-Hill, 1991.

[9] M. Tarrab and A. Feuer, "Convergence and performance analysis of the normalized LMS algorithm with uncorrelated Gaussian data," *IEEE Trans. Inform. Theory*, vol. IT-34, pp. 680–691, 1988.

[10] B. Widrow and M. A. Lehr, "30 years of adaptive neural networks: Perceptron, Madaline, and backpropagation," *Proc. IEEE*, vol. 78, no. 9, pp. 1415–1441, 1990.

On Competitive Learning

Lipo Wang

Abstract—**We derive learning rates such that all training patterns are equally important statistically and the learning outcome is independent of the order in which training patterns are presented, if the competitive neurons win the same sets of training patterns regardless the order of presentation. We show that under these schemes, the learning rules in the two different weight normalization approaches, i.e., the length-constraint and the sum-constraint, yield practically the same results, if the competitive neurons win the same sets of training patterns with both constraints. These theoretical results are illustrated with computer simulations.**

I. Introduction

Competitive learning has been widely studied and applied [1]. Remarkable artificial neural-network models developed using competitive learning include, for example, the von der Malsburg model [2], the adaptive resonance theory (ART) [3], the self-organizing map [4], the neocognitron [5], and the counterpropagation network ([1, p. 147]). Let us consider a simple competitive learning network [6] consisting of a layer of N_n competitive *neurons* and a layer of N input *nodes*, N being the dimension of input patterns. Suppose neuron k receives inputs from an input pattern $\vec{x}$ through a set of synaptic weights $\vec{w}_k$ connecting neuron k with all input nodes

$$\begin{aligned} h_k &= \sum_{j=1}^{N} w_{kj}x_j \\ &= \vec{w}_k \cdot \vec{x} \\ &= |\vec{w}_k| \cdot |\vec{x}| \cos\theta_k, \end{aligned} \tag{1}$$

where θ_k is the angle between the weight vector $\vec{w}_k$ and the input pattern $\vec{x}$. Suppose neuron i has the largest total input, i.e., $h_i = \max\{h_k, 1 \leq k \leq N_n\}$. Neuron i wins the competition and

Manuscript received January 23, 1996; revised March 26, 1997.

The author is with the School of Computing and Mathematics, Deakin University, Clayton, Victoria 3168, Australia.

Publisher Item Identifier S 1045-9227(97)05249-1.

becomes the sole neuron responding to the input pattern $\vec{x}$ in winner-take-all competitive learning. According to (1), if all weight vectors are normalized to have the same length

$$\begin{aligned}|\vec{w}_k|^2 &= \sum_{j=1}^{N} w_{kj}^2 \\ &= L^2, \qquad \text{for all } k \end{aligned} \tag{2}$$

it is the neuron whose weight vector is the most similar to the input pattern that wins the competition. During training, the winning neuron adjusts its weight vector to become more similar to the input pattern and this weight vector should then be renormalized

$$\vec{w}_i(\tau_i) = \frac{\vec{w}_i(\tau_i - 1) + \alpha(\tau_i)\vec{x}_i(\tau_i)}{|\vec{w}_i(\tau_i - 1) + \alpha(\tau_i)\vec{x}_i(\tau_i)|} \cdot L \tag{3}$$

where $\tau_i \geq 1$ is the number of times neuron i has modified its weights including the current update and $\alpha(\tau_i)$ is the learning rate at that time.

Prior to the existence of the *length-constraint* (2), von der Malsburg [2] proposed the following *sum-constraint* as a way to prevent unlimited growth of the synaptic weights (see also, e.g., [3], [6], and [7])

$$\sum_{j=1}^{N} w_{kj} = S, \qquad \text{for all } k. \tag{4}$$

The equivalence between the sum-constraint and the length-constraint is only approximate; however, the sum-constraint is believed to be biologically more plausible and is computationally simpler in comparison with the length-constraint. The sum-constraint (4) is automatically satisfied during weight adaptations if all training vectors are normalized according to the sum-constraint and

$$\vec{w}_i(\tau_i) = [1 - \alpha(\tau_i)]\, \vec{w}_i(\tau_i - 1) + \alpha(\tau_i)\vec{x}(\tau_i). \tag{5}$$

Recent work [8], [9] has criticized that the sum-constraint may not be a good approximation of the length-constraint and may lead to undesirable grouping of training patterns during learning [8].

Let us iterate (3) toward an earlier time step as shown in (6) at the bottom of the page. We observe from (6) that in general, exchanging $\vec{x}_i(\tau_i)$ with $\vec{x}_i(\tau_i - 1)$ will yield a different $\vec{w}_i(\tau_i)$, because the coefficients of $\vec{x}_i(\tau_i)$ and $\vec{x}_i(\tau_i - 1)$ may not be the same. Hence, the effect of each training pattern on the synaptic weights of the network, and therefore, the final outcome of learning, depends on the order in which the training patterns are presented, even if the competitive neurons win the same sets of training patterns regardless the order of presentation. In fact, if the learning rate α does not vary with time and is between zero and one [6], [8], (6) shows that the training pattern presented later $\vec{x}_i(\tau_i)$ carries a larger statistical weight compared with the training pattern presented earlier $\vec{x}_i(\tau_i - 1)$, because $\vec{w}_i(\tau_i - 2)$ is *usually* very similar to $\vec{x}_i(\tau_i - 1)$ and therefore, $L^{-1}|\vec{w}_i(\tau_i - 2) + \alpha(\tau_i - 1)\vec{x}_i(\tau_i - 1)| > L^{-1}|\vec{w}_i(\tau_i - 2)| = 1$. This effect may provide a "forgetting" mechanism for some on-line systems; however, it may not be desirable in other practical applications.

The condition under which the training outcome is *independent* of the order of training presentation and each training pattern has the same statistical importance can be readily obtained from (6)

$$\alpha(\tau_i) = \frac{\alpha(\tau_i - 1)L}{|\vec{w}_i(\tau_i - 2) + \alpha(\tau_i - 1)\vec{x}_i(\tau_i - 1)|}. \tag{7}$$

With (7) fulfilled for all τ_i and if the competitive neurons win the same sets of training patterns regardless the order of presentation, the weight vector under the length-constraint becomes a sum of all contributing training vectors, apart from a multiplicative constant and the weight initialization

$$\vec{w}_i(\tau_i) = \frac{\vec{w}_i(0) + \alpha(1)\displaystyle\sum_{\tau'=1}^{\tau_i} \vec{x}(\tau')}{\left|\vec{w}_i(0) + \alpha(1)\displaystyle\sum_{\tau'=1}^{\tau_i} \vec{x}(\tau')\right|} \cdot L. \tag{8}$$

Similar conclusions may be drawn for the sum-constraint, that is, the statistical importance of each training pattern again depend on the order in which training patterns are presented, and the following learning rate assures equal statistical importance of all training patterns and independence of presentation order in the case of the sum-constraint

$$\begin{aligned}\alpha(\tau_i) &= \frac{\alpha(\tau_i - 1)}{1 + \alpha(\tau_i - 1)} \\ &= \frac{\alpha(\tau_i - 2)}{1 + 2\alpha(\tau_i - 2)} = \cdots = \frac{\alpha(1)}{1 + (\tau_i - 1)\alpha(1)}\end{aligned} \tag{9}$$

if the competitive neurons win the same sets of training patterns regardless the order of presentation. Then the weight vector under the sum-constraint again becomes a sum of all contributing training vectors, apart from a multiplicative constant and the weight initialization

$$\begin{aligned}\vec{w}_i(\tau_i) = &\frac{1 - \alpha(1)}{1 + (\tau_i - 1)\alpha(1)}\, \vec{w}_i(0) \\ &+ \frac{\alpha(1)}{1 + (\tau_i - 1)\alpha(1)} \sum_{\tau'=1}^{\tau_i} \vec{x}(\tau').\end{aligned} \tag{10}$$

Thus, if we select learning rates according to (7) under the length-constraint or (9) under the sum-constraint and if the competitive neurons win the same sets of training patterns with both constraints, the end results of learning are practically the same in both cases: a sum of all contributing training patterns [(8) and (10)].

If we choose $\alpha(1) = 1$, the learning rates given (9) are simply $\alpha(\tau_i) = 1/\tau_i$, and the final weight vector does not depend on the weight initialization $\vec{w}_i(0)$ and becomes an *overall* average of the contributing training patterns, according to (10), $\vec{w}_i(\tau_i) = (1/\tau_i)\sum_{\tau'=1}^{\tau_i} \vec{x}(\tau')$. The results in this special case are the same as those derived by Wu and Fallside [10] for optimal vector quantization. Furthermore, Mulier and Cherkassky [11] have proposed learning rates which lead to equal weighting to all training patterns, for a system with neighborhood updates such as the self-organizing map.

The following simple computer experiments have been carried out to illustrate the above theoretical results. Ten images (11 × 11 pixels in size) shown in Fig. 1 were created as training patterns. Each black pixel in Fig. 1 represents a value 1.0 and each white pixel represents a value −1.0. Each pattern has 57 black pixels and 64 white pixels. These patterns satisfy both the sum-constraint [(4) with $S = 57 - 64 = -7$] and the length-constraint [(2) with $L = \sqrt{121} = 11$]. Three presentation sequences of the training patterns were used: $s_1 \equiv \{t_0 t_1 t_2 t_3 t_4 x_0 x_1 x_2 x_3 x_4\}$, $s_2 \equiv \{x_4 t_1 t_4 x_0 t_3 x_1 t_2 x_2 t_0 x_3\}$, and $s_3 \equiv \{t_2 x_3 x_4 t_0 t_4 x_0 x_1 t_3 x_2 t_1\}$. Four learning paradigms were used: 1) length-constraint (3) with a constant learning rate 0.5; 2) sum-constraint (5) with a constant learning rate 0.5; 3) length-constraint

$$\vec{w}_i(\tau_i) = \frac{\vec{w}_i(\tau_i - 2) + \alpha(\tau_i - 1)\vec{x}_i(\tau_i - 1) + L^{-1}|\vec{w}_i(\tau_i - 2) + \alpha(\tau_i - 1)\vec{x}_i(\tau_i - 1)|\alpha(\tau_i)\vec{x}_i(\tau_i)}{|\vec{w}_i(\tau_i - 2) + \alpha(\tau_i - 1)\vec{x}_i(\tau_i - 1) + L^{-1}|\vec{w}_i(\tau_i - 2) + \alpha(\tau_i - 1)\vec{x}_i(\tau_i - 1)|\alpha(\tau_i)\vec{x}_i(\tau_i)|} \cdot L. \tag{6}$$

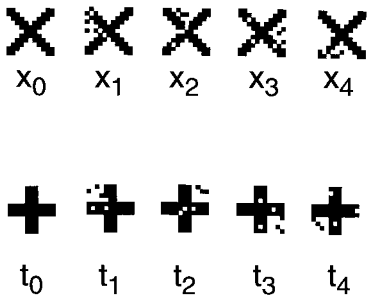

Fig. 1. Training patterns (11 × 11 pixels in size).

with learning rate given by (7) and $\alpha(1) = 1.0$; and 4) sum-constraint with learning rate given by (9) and $\alpha(1) = 1.0$. For each learning paradigm and each training sequence, we used a network of 11 × 11 input nodes and ten competitive neurons. Before training, we set 57 randomly selected synapses of each neuron to 1.0 and set the other 64 synapses to -1.0, thus both the sum- and the length-constraints were satisfied by the synapses. After training, the synapses of the *updated* neurons were displayed in Fig. 2. Three types of pixels (black if value is above 0.33, gray if between -0.33 and 0.33, white if below -0.33) are sufficient to illustrate the differences between the learning paradigms. The two assumptions under which our two claims are made were true in every case: one of the neurons won for patterns t_0, t_1, t_2, t_3, and t_4, whereas, another neuron won for patterns x_0, x_1, x_2, x_3, and x_4, although which the two winning neurons were depended on the training sequence used (e.g., with all four paradigms, the first neuron won the t's and the ninth neuron won the x's for sequence s_1, whereas, the fifth neuron won the t's and the fourth neuron won the x's for sequence s_2, and so on). Fig. 2 illustrates the two theoretical results under the assumptions: with the learning rates given by (7) and (9), the training outcome does not depends on the order in which the training patterns are presented, and the sum- and length-constraints lead to practically the same results [Fig. 2(c) and (d)]; however, with other learning rates, training results may vary significantly with the presentation order of training patterns or the type of weight normalization [Fig. 2(a) and (b)]. To clearly illustrate our theoretical results, we have generated the training patterns shown in Fig. 1 in such a way that the clusters in the training vectors are obvious.

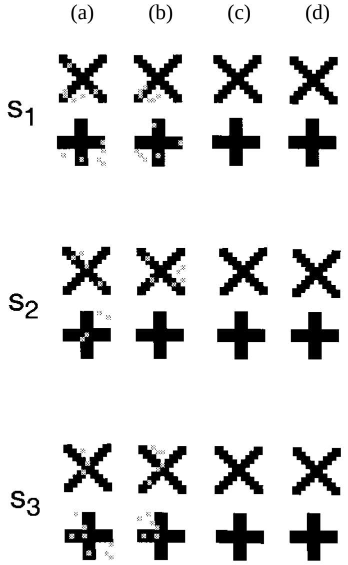

Fig. 2. Synapses of the winning neurons after training. Three presentation sequences of the training patterns shown in Fig. 1 were used: $s_1 \equiv \{t_0t_1t_2t_3t_4x_0x_1x_2x_3x_4\}$, $s_2 \equiv \{x_4t_1t_4x_0t_3x_1t_2x_2t_0x_3\}$, and $s_3 \equiv \{t_2x_3x_4t_0t_4x_0x_1t_3x_2t_1\}$. Four learning paradigms were used: (a) length-constraint 3 with a constant learning rate 0.5, (b) sum-constraint 5 with a constant learning rate 0.5, (c) length-constraint with learning rates given by (7) and $\alpha(1) = 1.0$, and (d) sum-constraint with learning rates given by (9) and $\alpha(1) = 1.0$.

It is worth noting that there are other ways of weighting the training vectors, in addition to equal weighting. For example, Andrew and Palaniswami [12] have shown that for a vector quantizer, slightly less weight should be given to the first few training patterns. Bauer *et al.* [13] have recently demonstrated that by adjusting the learning rates appropriately, the final distribution of the exemplars may be altered to model the input probability distribution function raised to an arbitrary power.

II. Conclusion

In summary, we have derived learning rates in competitive learning such that all training patterns have equal statistical importance and the final outcome of learning does not depend on the order in which the training patterns are presented, if the competitive neurons win the same sets of training patterns regardless the order of presentation. We have proven that once these learning rates are used, practically the same results are obtained with competitive learning algorithms based on the weight length- and sum-constraints, if the competitive neurons win the same sets of training patterns with both constraints. These theoretic results are demonstrated with some simple computer experiments. To what extend these assumptions may be relaxed and applications of these learning rates to practical data will be the subject of future studies.

Acknowledgment

The author thanks the reviewers for many helpful comments and suggestions, and for pointing out some important references.

References

[1] The earliest origin of the competitive learning rule has not been established. See discussions in R. Hecht-Nielsen, *Neurocomputing.* Reading, MA: Addison-Wesley, 1990, p. 63.

[2] C. von der Malsburg, "Self-organization of orientation sensitive cells in the striate cortex," *Kybernetik,* vol. 14, pp. 85–100, 1973.

[3] G. A. Carpenter and S. Grossberg, "Neural dynamics of category learning and recognition: Attention, memory consolidation, and amnesia," in *Brain Structure, Learning, and Memory,* J. Davis, R. Newburgh, and E. Wegman, Eds. Boulder, CO: Westview, (based on AAAS Symp. Series held in 1984), 1988, pp. 233–290.

[4] T. Kohonen, *Self-Organization and Associative Memory.* Berlin: Springer-Verlag, 1989.

[5] K. Fukushima, "Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position," *Biol. Cybern.,* vol. 36, pp. 193–202, 1980.

[6] D. E. Rumelhart and D. Zipser, "Feature discovery by competitive learning," *Cog. Sci.,* vol. 9, pp. 75–112, 1985.

[7] K. Grajski and M. Merzenich, "Hebb-type dynamics is sufficient to account for the inverse magnification rule in cortical somatotopy," *Neural Computa.,* vol. 2, pp. 71–84, 1990.

[8] G. G. Sutton, III and J. A. Reggia, "Effects of normalization constraints on competitive learning," *IEEE Trans. Neural Networks,* vol. 5, pp. 502–504, May 1994.

[9] G. J. Goodhill and H. G. Barrow, "The role of weight normalization in competitive learning," *Neural Computa.,* vol. 6, pp. 255–269, 1994.

[10] L. Wu and F. Fallside, "On the design of connectionist vector quantizers," *Comput. Speech and Language,* vol. 5, pp. 207–229, 1991.

[11] F. M. Mulier and V. S. Cherkassky, "Statistical analysis of self-organization," *Neural Networks,* vol. 8, no. 5, pp. 717–727, 1995.

[12] L. L. Andrew and M. Palaniswami, “A unified approach to selecting optimal step lengths for adaptive vector quantizers,” *IEEE Trans. Commun.*, vol. 44, pp. 434–439, 1996.

[13] H.-U. Bauer, R. Der, and M. Herrmann, “Controlling the magnification factor of self-organizing feature maps,” *Neural Computa.*, vol. 4, pp. 757–771, 1996.

Comments on “A Training Rule which Guarantees Finite-Region Stability for a Class of Closed-Loop Neural-Network Control Systems”

Sangbong Park and Cheol Hoon Park

Abstract—**In this letter, we show that the proof of Proposition 1 and the proposed stability condition as training constraints are not correct and therefore that the stability of the neural-network control system is not quite right. We suggest a modified version of the proposition with its proof and comment on another problem of the paper.**

I. Introduction

In the above paper,[1] a training method for a neural-network control system which guarantees local closed-loop stability is proposed based on a Lyapunov function and a modified standard backpropagation (SBP) training rule. Proposition 1 of the paper is as follows.

Proposition 1: The equilibrium state of the control system consisting of the system with neural-network controller is stable if there exists a real $(n \times n)$ positive symmetric definite matrix $\boldsymbol{P}$, a real $(n \times p)$ matrix q, and a real $(p \times p)$ matrix Γ such that

$$A^T PA - P = -qq^T - Q \tag{7a}$$

$$A^T PBW_2 + (W_1)^T = -q\Gamma^T \tag{7b}$$

and

$$\Gamma\Gamma^T + W_2^T B^T PBW_2 = 2I \tag{7c}$$

where Q is a real $(n \times n)$ positive symmetric matrix and I is the identity matrix of dimension p.

Furthermore,

$$V = x(k)^T Px(k) \tag{8}$$

is a discrete Lyapunov function for the control system.

In order to prove the Proposition 1, the authors use the Lyapunov candidate (8), and for all $k \geq 0$, after assuming the existence of P, q, and Γ, which satisfy (7). They obtain the rate of decrease of V as follows:

$$\begin{aligned} -dV &= V(k) - V(k+1) \\ &= x(k)^T Px(k) - x(k+1)^T Px(k+1). \end{aligned}$$

The proof of the proposition in the above paper showed stability of the neural-network control system by proving the existence of a Lyapunov function which satisfies the Lyapunov stability for existing weights. However, from (7) in Proposition 1, since the matrix P is highly dependent on weight matrices W_1 and W_2 and each weight matrix is iteratively obtained at each time step, the matrix P itself should be a function of time. The matrix P and other Q, q, Γ at time k are not the same with ones at time $k+1$. Therefore, the derivation of a Lyapunov candidate function and its proof are not right.

With the following modification, we prove the proposition.

The equilibrium state of the control system with neural-network controller is stable if there exist a real $(n \times p)$ matrix $q(k)$, a $(p \times p)$ matrix $\Gamma(k)$, and an $(n \times n)$ positive definite symmetric bounded matrix $P(k)$, that is, $0 < c_1 I \leq P(k) \leq c_2 I$ for all $k \geq 0$ [1] such that

$$\begin{aligned} &A^T P(k+1)A - P(k) \\ &\quad = -q(k+1)q(k+1) \\ &A^T P(k+1)BW_2(k+1) + W_1(k+1)^T \\ &\quad = -q(k+1)\Gamma(k+1)^T \\ &\Gamma(k+1)\Gamma(k+1)^T + W_2(k+1)^T B^T P(k+1)BW_2(k+1) \\ &\quad = 2I \qquad \text{for } k \geq 0 \end{aligned}$$

where $Q(k)$ is a symmetric positive definite matrix, that is, $Q(k) \geq c_3 I > 0$ for all $k \geq 0$.

Proof: The proof is straightforward [1], and similar to the proof in the above paper. We use

$$V = x(k)^T P(k)x(k) \tag{8}$$

Lyapunov function candidate of the control system. The rate of decrease of V is

$$\begin{aligned} -dV &= V(k) - V(k+1) \\ &= x(k)^T P(k)x(k) - x(k+1)^T P(k+1)x(k+1) \\ &\geq x(k)^T Q(k+1)x(k) - 2\gamma\|P\| \cdot \|x\|^2 \\ &\geq (c_3\|x\|^2 - 2\gamma\|P\| \cdot \|x\|^2) \\ &= (c_3 - 2c_2\gamma)\|x\|^2, \qquad \text{whenever } \|x\| \leq \delta. \end{aligned}$$

Choosing $\gamma < c_3/2c_2$ ensures that the rate of increase of V is negative definite in $\|x\| \leq \delta$, and therefore it is concluded that the equilibrium state of the control system is stable. Based on the above modification, the constraints of the training algorithm of the paper should be changed.

Another problem of the paper is that in order to guarantee the stability of the system the authors should show the finite convergence of the training algorithm at each stage, which is not considered in the paper. Moreover, the number of minor-iterative processes for which a solution should be found is problem dependent. So we think the training algorithm suggested in the paper may not be very useful in the real digital control system in spite of the above modification. Consequently, the stability of the control system and its performance with proposed algorithm in the above paper are still to be investigated.

References

[1] H. K. Khalil, *Nonlinear Systems.* New York: Macmillan, 1992.

Manuscript received September 10, 1996; revised January 8, 1997.

The authors are with the Department of Electrical Engineering, Korea Advanced Institute of Science and Technology, 373-1 Kusong-dong Yusong-gu, Taejon 305-701, Korea.

Publisher Item Identifier S 1045-9227(97)05250-8.

[1] S. Kuntanapreeda and R. R. Fullmer, *IEEE Trans. Neural Networks*, vol. 7, no. 3, pp. 745–751, 1996.