

“© 2011 IEEE. Personal use of this material is permitted. Permission from IEEE must be obtained for all other uses, in any current or future media, including reprinting/republishing this material for advertising or promotional purposes, creating new collective works, for resale or redistribution to servers or lists, or reuse of any copyrighted component of this work in other works.”

Robust online adaptive neural network control for the regulation of treadmill exercises

Tuan Nghia Nguyen, *Member, IEEE*, Hung Nguyen, *Senior Member, IEEE* and Steven Su, *Member, IEEE*, Branko Celler, *Member, IEEE*

Abstract—The paper proposes a robust online adaptive neural network control scheme for an automated treadmill system. The proposed control scheme is based on Feedback-Error Learning Approach (FELA), by using which the plant Jacobian calculation problem is avoided. Modification of the learning algorithm is proposed to solve the overtraining issue, guaranteeing to system stability and system convergence. As an adaptive neural network controller can adapt itself to deal with system uncertainties and external disturbances, this scheme is very suitable for treadmill exercise regulation when the model of the exerciser is unknown or inaccurate. In this study, exercise intensity (measured by heart rate) is regulated by simultaneously manipulating both treadmill speed and gradient in order to achieve fast tracking for which a single input multi output (SIMO) adaptive neural network controller has been designed. Real-time experiment result confirms that robust performance for nonlinear multivariable system under model uncertainties and unknown external disturbances can indeed be achieved.

I. INTRODUCTION

Regular exercises and physical activities are beneficial for long-term health and well-being. This study investigates the regulation of treadmill exercises by feedback control of exercisers' heart rate. For the regulation of heart rate during exercises, there are several existing methods in literature. For example, in [1], a support vector machine (SVM) based nonlinear control approach has been developed. Paper [2] developed a nonlinear Model Predictive Controller for safe rehabilitation exercises. However, the above mentioned methodologies are all using SISO controller for which the manipulated variable is treadmill speed. In order to reach the subject's limit of tolerance for exercise testing and rehabilitation of subjects with impaired exercise tolerance, ramp type protocols were proposed by simultaneously manipulating both speed and gradient (without feedback) in [3, 4]. In [5], a multi-loop PID controller based HR tracking system has been presented which independently adjusts both treadmill speed and gradient in closed loop, and achieved good performance. However, the multi-loop PID controller designed in [5] may need re-tuning of parameters for different exercisers, which

generates extra supervision task for individual exercises. In order to overcome this deficiency, this study developed an adaptive neural network controller, which is capable of self-tuning controller parameters based on its learning ability.

Neural networks are equipped with many desirable properties, including learning by experience, the ability to map nonlinear functions, the ability to generalize, as well as demonstrating robustness with noise and multivariable interactions. As such, neural networks have provided effective solutions for complex and nonlinear control problems, both with and without uncertainties [6-8]. In most neural network control structures, a neural network controller is connected serially to an unknown plant. It therefore faces the problem of unknown plant Jacobian, which is defined as the partial derivative of a plant's outputs with respect to its inputs when propagating the control error using back propagation method to the adjustable coefficients of the neural controller. This adjustment leads to error reduction.

Due to the complexity of an unknown plant, it is difficult or even impossible to calculate a plant Jacobian. Several methods have been proposed for finding a plant Jacobian, though none have proven effectiveness. These methods include the online approximation method [9], the plant Jacobian sign calculation [10], direct calculation from the plant model [11], the Inverse Transfer Matrix scheme [6], gain layer schemes [12, 13], and the use of trained neural networks as identifier [11]. To avoid the calculation of a plant Jacobian, an adaptive updating law has been proposed by Fierro and Lewis [14]. This adaptive updating law is based on Lyapunov theory. Although this method guarantees the stability of the controlled system, the proposed updating rules are too complicated for a real-time system.

To solve the plant Jacobian problem, the robust neuro-sliding mode multivariable control strategy is proposed by T.N. Nguyen (an author of this article) in [15]. This control strategy effectively deals with multivariable control problem based on the combination of decoupling technique and neuro-sliding mode control. However, this control strategy requires that the number of inputs and outputs of a multivariable system are the same.

Another solution to deal with plant Jacobian problem is to utilize Feedback Error Learning Approach (FELA) for neural network control design [16, 17]. Several works using neural network control scheme based FELA for control applications have been published [18-20]. However, none of these articles discuss on the overtraining problem issue, which may lead to system unstable condition.

Tuan Nghia Nguyen is with Faculty of Engineering, University of Technology, Sydney, Broadway, NSW 2007, Australia (phone: +612-9514-2995; fax: +61 2 9514 2868; e-mail: TuanNghia.Nguyen@uts.edu.au).

Hung .T Nguyen is with Faculty of Engineering, University of Technology, Sydney, Broadway, NSW 2007, Australia (e-mail: Hung.Nguyen@uts.edu.au).

Steven Su is with Faculty of Engineering, University of Technology, Sydney, Broadway, NSW 2007, Australia (e-mail: Steven.Su@uts.edu.au).

Branko Celler is with the College of Health and Science, University of Western Sydney. (e-mail: b.cellier@uws.edu.au).

In this paper, we propose a robust online adaptive neural network control scheme for a multivariable control problem, and this proposed scheme can be applicable for a multiple input single output system. The proposed control scheme is based on FELA, thus the calculation of plant Jacobian is not required. A modification of learning algorithm is proposed to solve the overtraining problem, thus guarantees the system stability and convergence. The effectiveness of the proposed control scheme is proved via real-time control of the automated treadmill system.

The paper is organized as follows. In Section II, Robust on-line adaptive neural network control scheme is presented in detail. Real-time experimental results for treadmill control system and discussions are shown in Section III. The conclusion is given in Section IV.

II. ROBUST ON-LINE ADAPTIVE NEURAL NETWORK CONTROL

The on-line adaptive neural network control scheme is shown in *figure 2.1*. The objective is to train the neural network controller online so that the output of the unknown plant (y) can track the reference profile (y_d) in shortest period of time (T_0). In this control scheme, the neural network controller is Multilayer-FeedForward neural network, where its structure is illustrated in *figure 2.2*. In order to be suitable for dynamics control purpose, first derivative and second derivative of reference y_d and error (e) are selected as neural network inputs [19].

For a given neural network structure, the output of the neural network can be calculated by using the following formula:

$$u_{NN} = f_2[\bar{W} \cdot f_1(W \cdot p + b) + \bar{b}] \quad (2.1)$$

where, f_1 and f_2 are activation function of the input-hidden layer and hidden-output layer, defined in equations (2.2) and (2.3), $W \in R^{S \times R}$, $\bar{W} \in R^{Q \times S}$, $b \in R^{S \times 1}$, $\bar{b} \in R^{Q \times 1}$, $p \in R^{R \times 1}$, and $u_{NN} \in R^{Q \times 1}$, and R is number of input nodes, S is number of hidden nodes, and Q is number of output nodes.

$$f_1(n) = \frac{1 - e^{-2n}}{1 + e^{-2n}} \quad (2.2)$$

$$f_2(n) = n \quad (2.3)$$

The cost function, therefore, is defined as the following equation:

$$J = \frac{1}{2} [y_d - y]^2 = \frac{1}{2} [e]^2 \quad (2.4)$$

where, $y \in R^{Z \times 1}$, $y_d \in R^{Z \times 1}$, $e \in R^{Z \times 1}$. Z is number of the unknown plant's output.

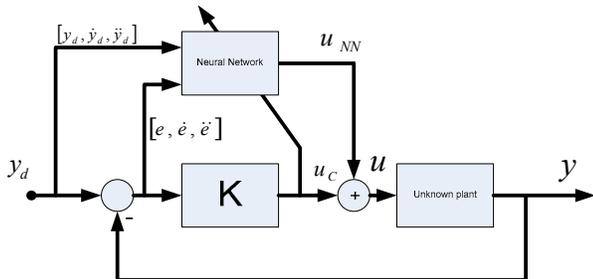


Figure 2.1: Online-adaptive neural network control scheme

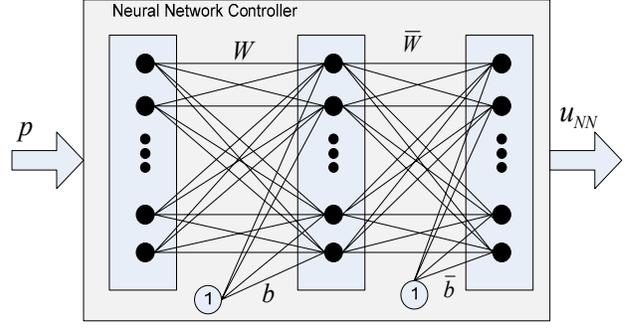


Figure 2.2: Neural Network Control structure

Based on the control scheme in the *figure 2.1*, the following formulas are obtained:

$$u_c = K \cdot e = u - u_{NN} \quad (2.5)$$

Multiple H with both sides of the equation (2.5), following formula is obtained:

$$H \cdot K \cdot e = H(u - u_{NN}) \quad (2.6)$$

where $K \in R^{Q \times Z}$, $H \in R^{Z \times Q}$.

Define: $T = H \cdot K$, where $T \in R^{Z \times Z}$.

Thus,

$$e = T^{-1} \cdot H \cdot (u - u_{NN}) \quad (2.7)$$

To minimize the cost function J , it is necessary to change the weights of the neural network controller to the direction of a negative gradient. Applying the chain rule, the results are obtained as follows:

$$\Delta W = \frac{\partial J}{\partial W} = \frac{\partial J}{\partial e} \frac{\partial e}{\partial u_{NN}} \frac{\partial u_{NN}}{\partial W} = -e \cdot T^{-1} \cdot H \cdot \frac{\partial u_{NN}}{\partial W} \quad (2.8)$$

$$\Delta b = \frac{\partial J}{\partial b} = \frac{\partial J}{\partial e} \frac{\partial e}{\partial u_{NN}} \frac{\partial u_{NN}}{\partial b} = -e \cdot T^{-1} \cdot H \cdot \frac{\partial u_{NN}}{\partial b} \quad (2.9)$$

$$\Delta \bar{W} = \frac{\partial J}{\partial \bar{W}} = \frac{\partial J}{\partial e} \frac{\partial e}{\partial u_{NN}} \frac{\partial u_{NN}}{\partial \bar{W}} = -e \cdot T^{-1} \cdot H \cdot \frac{\partial u_{NN}}{\partial \bar{W}} \quad (2.10)$$

$$\Delta \bar{b} = \frac{\partial J}{\partial \bar{b}} = \frac{\partial J}{\partial e} \frac{\partial e}{\partial u_{NN}} \frac{\partial u_{NN}}{\partial \bar{b}} = -e \cdot T^{-1} \cdot H \cdot \frac{\partial u_{NN}}{\partial \bar{b}} \quad (2.11)$$

From equation (2.1), the third terms in the equations (2.8)-(2.11) can be easily obtained as follows:

$$\frac{\partial u_{NN}}{\partial W} = f_2'[\bar{W} \cdot f_1(W \cdot p + b)] \cdot f_1'(Wp + b) \bar{W} \cdot p \quad (2.12)$$

$$\frac{\partial u_{NN}}{\partial b} = f_2'[\bar{W} \cdot f_1(W \cdot p + b)] \cdot f_1'(Wp + b) \bar{W} \quad (2.13)$$

$$\frac{\partial u_{NN}}{\partial \bar{W}} = f_2'[\bar{W} \cdot f_1(W \cdot p + b)] \cdot f_1(Wp + b) \quad (2.14)$$

$$\frac{\partial u_{NN}}{\partial \bar{b}} = f_2'[\bar{W} \cdot f_1(W \cdot p + b)] \quad (2.15)$$

Finally, the updating rules for neural network controller are in the following formulas:

$$W_{i,j} = \begin{cases} W_{i,j} - \mu \cdot \Delta W_{i,j}, & |\Delta W_{i,j}| < 30\% \cdot W_{i,j} \\ W_{i,j}, & \text{otherwise} \end{cases} \quad (2.16)$$

$$b_i = \begin{cases} b_i - \mu \cdot \Delta b_i, & |\Delta b_i| < 30\% \cdot b_i \\ b_i, & \text{otherwise} \end{cases} \quad (2.17)$$

$$\bar{W}_{i,j} = \begin{cases} \bar{W}_{i,j} - \mu \cdot \Delta \bar{W}_{i,j}, & |\Delta \bar{W}_{i,j}| < 30\% \cdot \bar{W}_{i,j} \\ \bar{W}_{i,j}, & \text{otherwise} \end{cases} \quad (2.18)$$

$$\bar{b}_i = \begin{cases} \bar{b}_i - \mu \cdot \Delta \bar{b}_i, & |\Delta \bar{b}_i| < 30\% \cdot \bar{b}_i \\ \bar{b}_i, & \text{otherwise} \end{cases} \quad (2.19)$$

where, μ is learning rate, which is chosen relatively small, $W_{i,j}, \bar{W}_{i,j}$ are the element (i,j) of the respecting weight matrices and b_i, \bar{b}_i are the element (i) of the weight vectors of the neural network controller.

These proposed updating rules will prevent a large change in the direction of a negative gradient, thus in turn over training problem will be avoided and prevented.

To guarantee that neural network controller can drive the plant outputs to track their references in desirable time T_0 , an off-line training scheme may be applied. The purpose of the off-line training is to find relatively optimal weights for the neural controller which guarantees fast convergence on the on-line control scheme in *figure 2.1*.

III. REAL-TIME EXPERIMENTAL RESULTS AND DISCUSSIONS

Hardware setup:

The block diagram of the treadmill control system is shown in *figure 3.1*. A subject's heart-rate is measured and transferred to the computer system via Bluetooth technology with sampling rate as 300 [samples/second]. In the computer system, online adaptive neural network controller is implemented in LabVIEW software using equations from (2.1) to (2.19). The sampling rate in the software is chosen as 50 [ms]. The treadmill interfaces with the computer system via R232 communication. A special protocol for the treadmill is provided by the treadmill provider (*TrackMaster*) so that the computer system can control treadmill speed and its gradient via RS232 port.

As the treadmill system consists of two inputs (velocity and gradient) and one output (heart rate), the neural network structure is (6,3,2) which is equivalent to 6 input nodes, 3 hidden nodes, and 2 output nodes. The hidden nodes are chosen by trial and error to obtain optimal performance of the control system. The learning rate (μ) is chosen as 0.0001, the gain K and H are in equation (3.1).

$$K = \begin{bmatrix} 0.5 \\ 0.5 \end{bmatrix}; H = \begin{bmatrix} 1 & 1 \end{bmatrix} \quad (3.1)$$

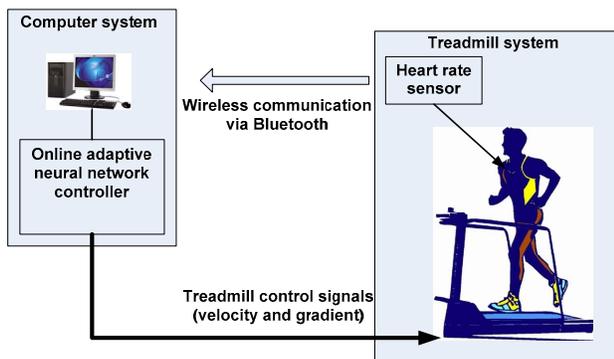


Figure 3.1: The block diagram of treadmill control system

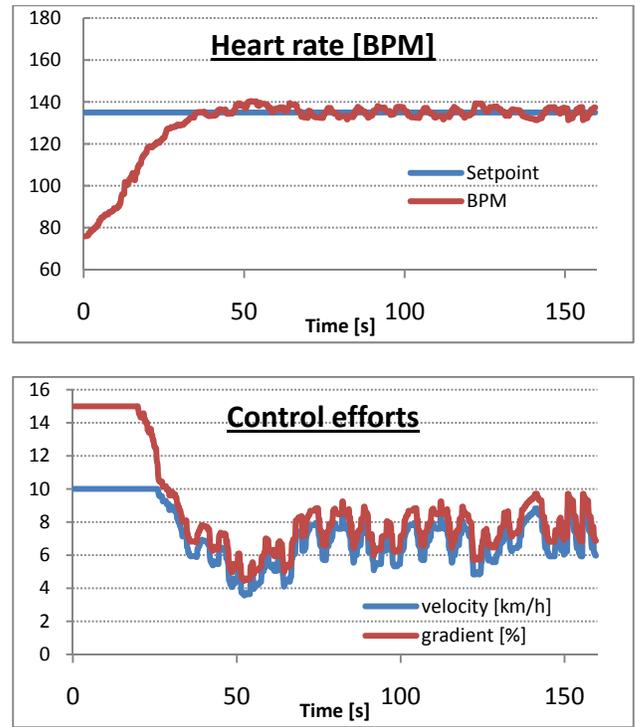


Figure 3.2: Heart rate tracking result, set-point is 135[bpm]

Experiment 1:

This experiment tests the performance of the treadmill control system with the proposed robust on-line adaptive neural network controller. A subject aged 29 was participated in this experiment. The desired heart rate set-point was set to be 135 beats per minute (*bpm*). The objective is to control both speed and gradient of the treadmill so that the subject heart-rate can track the set-point in a desirable time ($T_0 \leq 100$ [s]).

Discussion:

The experiment result is shown in *figure 3.2*. As seen in the obtained result, the performance of the treadmill system controlled by the proposed neural network controller is significant improved compared to the MISO method presented in [5]. The output response is obtained with no overshoot, and setting time $T_0 = 38$ [s]. The advantage of the proposed method is that no plant model is required as the adaptive neural network controller is learnt online to adaptive with system dynamics changes. As a result, the neural network controller can deal with system uncertainties and reject external disturbances.

Experiment 2:

This experiment tests the adaptation ability of the online adaptive neural network controller when system parameters are changed. Four subjects aged from 29 to 34 year olds were participated in this experiment as shown in table 3.1. The set-point heart rate was set to be 145[bpm]. The objective of this experiment is to maintain the system performance with a different set-point and different subjects involved in the exercise.

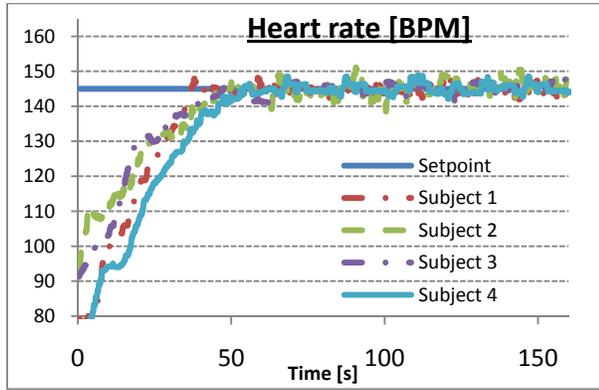


Figure 3.3: Heart rate tracking results of four subjects, set-point is 145[bpm]

Subject ID	Age	Height (cm)	Weight (kg)
Subject 1	29	165	62
Subject 2	31	170	68
Subject 3	33	168	65
Subject 4	34	180	85

Table 3.1: Participated subjects' characteristics

Heart-rate Set-point	Overshoot %	Rise time [s]	Settling time[s]
135 [bpm]	0	27	38
145 [bpm]	0	34	48

Table 3.2: System performance with different set-points for subject 1 (age 29), according to Figure 3.2 and Figure 3.3 results.

Discussion:

The results of the four subjects' heart rate tracking were shown in *figure 3.3*. Clearly, the robust online adaptive neural network controller still guarantees the system performance with no overshoot and settling time $T_0 < 50$ [s] regardless of different testing conditions and different plant models (Table 3.1 and 3.2). This result proves the effectiveness the proposed control scheme even under condition of system uncertainties and external disturbances.

Note: In these real-time experiments, online adaptation scheme is applied directly without requiring offline-training scheme. For other systems which require faster dynamics response, the offline training scheme may be applied.

IV. CONCLUSION

This study explores the regulation of treadmill exercisers by simultaneously adjusting treadmill speed and gradient. We have proposed a robust online adaptive neural network control scheme for unknown multivariable systems based on Feedback-Error Learning Approach. In the proposed control scheme, modified learning algorithm for an adaptive neural network controller is also proposed to guarantee system stability and performance. To obtain fast convergence, off-line training scheme can be introduced to obtain optimal weights before on-line adaptation scheme is used. Real-time implementation of the proposed control scheme on the automated treadmill system has been conducted to prove effectiveness of this scheme. Obtained results of real time experiments show that the proposed control scheme

guarantees desirable system performance under model uncertainty and external disturbance conditions.

REFERENCES

- [1] S. W. Su, *et al.*, "Identification and Control for Heart Rate Regulation during Treadmill Exercise," *IEEE Trans on Biomedical Engineering*, vol. 54, pp. 1238-1246, 2007.
- [2] S. W. Su, *et al.*, "Optimizing heart rate regulation for safe exercise," *Annals of Biomedical Engineering*, vol. 38, pp. 758-768, 2010.
- [3] L. P. Jamieson, *et al.*, "A Treadmill Control Protocol Combining Nonlinear, Equally Smooth Increases in Speed and Gradient," *Medical Engineering & Physics*, vol. 30, pp. 747-754, 2008.
- [4] K. J. Hunt, "Treadmill Control Protocols for Arbitrary Work Rate Profiles Combining Simultaneous Nonlinear Changes in Speed and Angle," *Biomedical Signal Processing and Control*, vol. 3, pp. 278-282, 2008.
- [5] W. Kaili, *et al.*, "Fast tracking of a given heart rate profile in treadmill exercise," in *Engineering in Medicine and Biology Society (EMBC), 2010 Annual International Conference of the IEEE*, 2010, pp. 2569-2572.
- [6] V. C. Chen and Y.-H. Pao, "Learning control with neural networks," in *IEEE International Conference on Robotics and Automation*, 1989, pp. 1448 - 1453
- [7] K. S. Narendra and K. Parthasarathy, "Identification and control of dynamical systems using neural networks," *IEEE Trans on Neural Networks*, vol. 1, pp. 4--27, 1990.
- [8] D. Psaltis, *et al.*, "A multilayered neural network controller," *IEEE Control Systems Magazine*, vol. 8, pp. 17-21, 1988.
- [9] L. Acosta, *et al.*, "On the Design and Implementation of a Neuromorphic Self-Tuning Controller," *Neural Process. Lett.*, vol. 9, pp. 229-242, 1999.
- [10] M. Saerens and A. Soquet, "Neural controller based on back-propagation algorithm," in *IEE Proceedings F on Radar and Signal Processing*, 1991, pp. 55-62.
- [11] G. W. Ng and P. A. Cook, "On-line adaptive control of non-linear plants using neural networks with application to liquid level control," *International Journal of Adaptive Control and Signal Processing*, vol. 12, pp. 13 - 28, 1998.
- [12] S. Zuo, *et al.*, "Robust motion control for nonholonomic constrained mechanical systems: sliding mode approach," in *American Control Conference, 2005. Proceedings of the 2005*, 2005, pp. 2883-2888 vol. 4.
- [13] K. P. Venugopal, *et al.*, "An improved scheme for direct adaptive control of dynamical systems using backpropagation neural networks," *Circuits, Systems, and Signal Processing*, vol. 14, pp. 213 - 236, 1995.
- [14] R. Fierro and F. L. Lewis, "Control of a nonholonomic mobile robot using neural networks," *Neural Networks, IEEE Transactions on*, vol. 9, pp. 589-600, 1998.
- [15] N. Tuan Nghia, *et al.*, "Robust Neuro-Sliding Mode Multivariable Control Strategy for Powered Wheelchairs," *Neural Systems and Rehabilitation Engineering, IEEE Transactions on*, vol. 19, pp. 105-111, 2011.
- [16] K. M. Kawato, R. Suzuki, "A Hierarchical Neural Network-Model for Control and Learning of Voluntary Movement," *Biological Cybernetics*, vol. 57, pp. 169-185, 1987.
- [17] M. Kawato and H. Gomi, "A computation model for four regions of the cerebellum based on feedback-error learning," *Biological Cybernetics*, vol. 68, pp. 95-103, 1992.
- [18] H. Miyamoto, *et al.*, "Feedback-Error-Learning Neural Network for Trajectory Control of a Robotic Manipulator," *Neural Networks*, vol. 1, pp. 251-265, 1998.
- [19] M. Teshnehlab and K. wantanabe, "Neural Network Controller with Flexible structure based on Feedback-Error-Learning Approach " *Journal of Intelligent and Robotic Systems*, vol. 15, pp. 367-387, 1996.
- [20] V. D. Kalanovic, *et al.*, "Feedback error learning neural network for trans-femoral prosthesis," *Rehabilitation Engineering, IEEE Transactions on*, vol. 8, pp. 71-80, 2000.