31st Annual International Conference of the IEEE EMBS
Minneapolis, Minnesota, USA, September 2-6, 2009

Radial Basis Function Neural Network-based PID Model for Functional Electrical Stimulation System Control

Longlong Cheng, Guangju Zhang, Baikun Wan, Linlin Hao, Hongzhi Qi and Dong Ming

Abstract—**Functional electrical stimulation (FES) has been widely used in the area of neural engineering. It utilizes electrical current to activate nerves innervating extremities affected by paralysis. An effective combination of a traditional PID controller and a neural network, being capable of nonlinear expression and adaptive learning property, supply a more reliable approach to construct FES controller that help the paraplegia complete the action they want. A FES system tuned by Radial Basis Function (RBF) Neural Network-based Proportional–Integral–Derivative (PID) model was designed to control the knee joint according to the desired trajectory through stimulation of lower limbs muscles in this paper. Experiment result shows that the FES system with RBF Neural Network-based PID model get a better performance when tracking the preset trajectory of knee angle comparing with the system adjusted by Ziegler- Nichols tuning PID model.**

I. Introduction

Functional electrical stimulation (FES) is an advancing technology for restoring paralyzed motor functions caused by a spinal cord injury or a stroke. It applies programmed electrical stimuli to intact peripheral nerves or muscles to help the quadriplegic patients completing the motion they want. It is generally accepted that FES needs better control strategies to extend the clinical application and how to modulate the parameters of the system to make the patients achieve the desired motion satisfactorily is one of the most important issues in FES application [1].

Most FES systems in early stage adopted open-loop control strategies [2,3] which is even preferred currently by many therapist as its simpleness although initial and periodical adjustment were needed in stimulation patterns for individual patients. However, the performance of the open-loop controllers is not satisfying owing to disturbances from external loads and muscle fatigue. Subsequently, closed-loop controller [4-6] was introduced to adjust the parameters of the FES system by feedback algorithm according to the error between the actual and desired output, which enhanced the precision and stability of the system significantly. Proportional–Integral–Derivative (PID) controller is a generic control loop feedback mechanism that widely used in industrial control systems. It attempts to correct the error between a measured variable and a desired setpoint through the proportional (P), integral (I) and derivative (D) values, and then keeps the error minimal. Veltink et al [7], in 1992, used the PID in FES system to control the knee joint angle. How to fix the proportional, integral and derivative values for PID controller is very important, especially in FES application which expects an exigent exactitude and stability. The traditional estimation of the P, I and D value for conventional PID is base on the industrial experience and manual tuning, such as Ziegler–Nichols method. In order to improve the control effect, some adaptive algorithms were developed. For example, Visioli A et al [8] shown a fuzzy logic-tuned PID controller; Benaskeur, A.R et al [9] illustrated a backstepping-based adaptive PID control and Yasue Mitsukura et al [10] used genetic algorithm to fix the parameters of PID.

This study adopts a Radial Basis Function (RBF) Neural Network-based PID scheme into FES strategy. The proportional, integral and derivative values of the PID are optimized through the self-training RBF Neural Network, and then used to control the knee joint angle by adjusting the pulse intensity in FES system. The result is compared, in this paper, with the controller modulated by Ziegler-Nichols which is widely used in Functional Electrical Stimulation application currently.

II. Methods

1. PID Algorithm

The discrete-form of the PID algorithm, with input error (t) and output u (t) is generally given as

$$u(t) = K_p error(t) + K_i \sum_{j=0}^{t} error(j) + K_d[error(t) - error(t-1)] \quad (1)$$

where K_p is the proportional value, K_i is the integral value, and K_d is the derivative value. *error (t)* means the difference

Manuscript received April 20, 2009. This research was supported by The National High Technology Research and Development Program of China (No. 2007AA04Z236), National Natural Science Foundation of China (No. 60501005), Biomedical Engineering Key Program (No. 07ZCKFSF01300) and Intentional Cooperation Key Program (No.08ZCGHHZ00300) of Tianjin Science Technology Support Plan.

Longlong Cheng, Guangju Zhang, Baikun Wan, Hongzhi Qi and Dong Ming are with Department of Biomedical Engineering, College of Precision Instruments and Optoelectronics Engineering, Tianjin University, Tianjin 300072, P. R. China (corresponding author: Prof Dong Ming, phone: 0086-22-27408718; fax: 0086-22-27406726; e-mail: richardming@tju.edu.cn).

Linlin Hao is with Jiangdulu Hospital, Tianjin 300250, P. R. China

Longlong Cheng is also with Department of Biomedical engineering, Case Western Reserve University, Cleveland OH 44106, USA.

between desired and measured output of controlled subject.

2. Ziegler–Nichols PID

Ziegler–Nichols, introduced by John G. Ziegler and Nathaniel B. Nichols, is almost the most popular method currently in FES application.

It confirms the PID parameters by escalating the K_p until the system starts to oscillate while $K_i = 0, K_d = 0$, and then calculates the PID parameters by following formulas:

$$K_p = 0.6K_p^{'} \tag{2}$$

$$K_d = \frac{K_p \pi}{4\omega} \tag{3}$$

$$K_i = \frac{K_p \omega}{\pi} \tag{4}$$

where $K_p^{'}$ is the proportional value when the oscillation appearances and ω is the circular frequency gotten from $\omega = \theta / T$ (θ is the position of the pole in the unit circle and *1/T* is the sampling rate)

3. PID Model based on RBF Neural Network

A typical RBF, which is a three-layer feed-forward network configuration, is shown in Fig.1. The neurons in the hidden layer contain Gaussian transfer functions whose outputs are inversely proportional to the distance from the center of the neuron. A nonlinear mapping from input to output, while linear from hidden layer to output layer in RBF Neural Network can enhance the learning rate and avoid a local minimum.

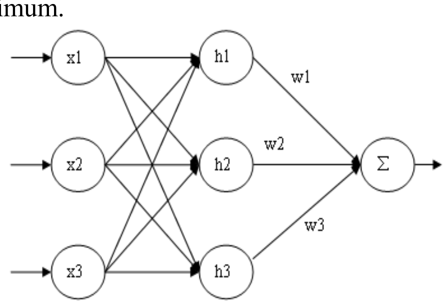

Fig.1 RBF neural network

3.1 Identification algorithm of controlled plant

Suppose that the Radial Basis Vector

$$H = [h_1, h_2, ...h_m]^T.$$

where h_j is the static Gaussian function as the nonlinearity for the hidden layer processing elements.

$$h_j = \exp(-\frac{\left\| X - C_j \right\|^2}{2b_j^2}) \quad (j = 1, 2...., m) \tag{5}$$

the b_j stands for the standard deviation of the Gaussian, and the C_j is the centre value.

The weight vector for the network is *W* and

$$W = [w_1, w_2......w_m]^T$$

The network output y can be expressed as:

$$y_m(k) = w_1 h_1 + w_2 h_2 + ... + w_m h_m \tag{6}$$

The performance function

$$J = \frac{1}{2}(yout(k) - y_m(k))^2 \tag{7}$$

According to gradient descent algorithm, weights iterative algorithm, node center and radial parameters are as follows:

$$w_j(k) = w_j(k-1) + \eta(yout(k) - y_m(k))h_j + \alpha(w_j(k-1) - w_j(k-2)) \tag{8}$$

$$\Delta b_j = (yout(k) - y_m(k)) w_j h_j \frac{\left\| X - C_j \right\|^2}{b_j^3} \tag{9}$$

$$b_j(k) = b_j(k-1) + \eta \Delta b_j + \alpha(b_j(k-1) - b_j(k-2)) \quad .... (10)$$

$$\Delta c_{ji} = (yout(k) - y_m(k)) w_j \frac{x_j - c_{ji}}{b_j^2} \tag{11}$$

$$c_{ji}(k) = c_{ji}(k-1) + \eta \Delta c_{ji} + \alpha(c_{ji}(k-1) - c_{ji}(k-2)) \quad .... (12)$$

where η is the learning rate and α is the momentum gene.

Jacobian matrix (sensitivity of plant output to controlled input) algorithm can be shown as

$$\frac{\partial y(k)}{\partial \Delta u(k)} \approx \frac{\partial y_m(k)}{\partial \Delta u(k)} = \sum_{j=1}^{m} w_j h_j \frac{c_{ji} - x_j}{b_j^2} \tag{13}$$

where $x_j = \Delta u(k)$

3.2 PID parameters

The (1) can be shown as:

$$u(t-1) = K_p error(t-1) + K_i \sum_{j=0}^{t-1} error(j) + K_d [error(t-1) - error(t-2)] \cdots \quad .... (14)$$

$$\Delta u(t) = u(t) - u(t-1) = K_p(error(t) - error(t-1)) + K_i error(t) + K_d(error(t) - 2error(t-1) + error(t-2)) \quad .... (15)$$

According to (15)

$$u(t) = \Delta u(t) + u(t-1) =$$
$$u(t-1) + K_p(error(t) - error(t-1)) + K_i error(t)$$
$$+K_d(error(t) - 2error(t-1) + error(t-2))$$
$$\ldots\ldots\ldots\ldots\ldots\ldots\ldots \quad (16)$$

And *error(k)=rin(k)-yout(k)*

The three inputs of the PID are:

$$xc(1) = error(k) - error(k-1)$$
$$xc(2) = error(k)$$
$$xc(3) = error(k) - 2error(k-1) + error(k-2)$$
$$\ldots\ldots\ldots\ldots\ldots\ldots\ldots \quad (17)$$

The error target function is defined by

$$E(k) = \frac{1}{2} error^2(k)$$

The three parameters of the PID were modulated by gradient descent algorithm

$$\Delta k_p = -\eta \frac{\partial E}{\partial k_p} = -\eta \frac{\partial E}{\partial y} \frac{\partial y}{\partial \Delta u} \frac{\partial \Delta u}{\partial k_p}$$
$$= \eta error(k) \frac{\partial y}{\partial \Delta u} xc(1) \quad (18)$$

$$\Delta k_I = -\eta \frac{\partial E}{\partial k_I} = -\eta \frac{\partial E}{\partial y} \frac{\partial y}{\partial \Delta u} \frac{\partial \Delta u}{\partial k_I}$$
$$= \eta error(k) \frac{\partial y}{\partial \Delta u} xc(2) \quad (19)$$

$$\Delta k_D = -\eta \frac{\partial E}{\partial k_D} = -\eta \frac{\partial E}{\partial y} \frac{\partial y}{\partial \Delta u} \frac{\partial \Delta u}{\partial k_D}$$
$$= \eta error(k) \frac{\partial y}{\partial \Delta u} xc(3) \quad (20)$$

($\frac{\partial y}{\partial \Delta u}$ was gained from Jacobian matrix)

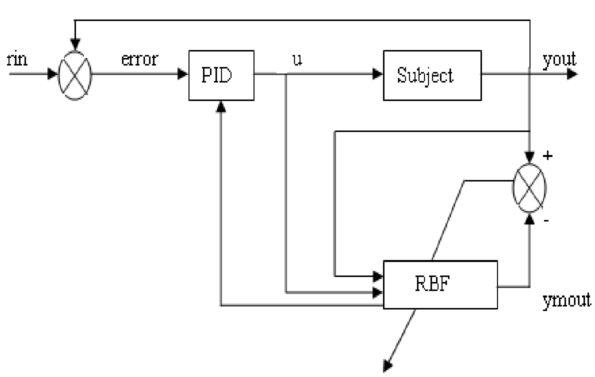

Fig.2 the structure of the PID modulated by RBF

The structure of PID base on the RBF is shown in the Fig.2. The *rin* in the figure is the desired trajectory and *yout* means measured output. The inputs of the Neural Network are $\Delta u(t)$, $yout(t)$ and $yout(t-1)$, and the PID parameters were modulated through the result of RBF Neural Network identification.

III. EXPERIMENT AND RESULTS

The Parastep produced by SIGMEDICS US was employed to study the knee joint control here. It is a non-invasive system with six stimulus channels and includes a microcomputer to generate the pulses. Five able-bodies, three males and two females, participated in this experiment. They sit calmly on the platform with relaxed calf and the knee extensors (quadriceps muscle group) were stimulated by a pair of surface electrodes. The cathode was placed on the motor point of rectus femoris and the anode was placed distally at the quadriceps tendon. The knee joint angle was controlled by changing the amplitude of the stimulation pulse. Defining the original knee angle was $\theta = 0$, when the lower leg was at rest during knee flexion.

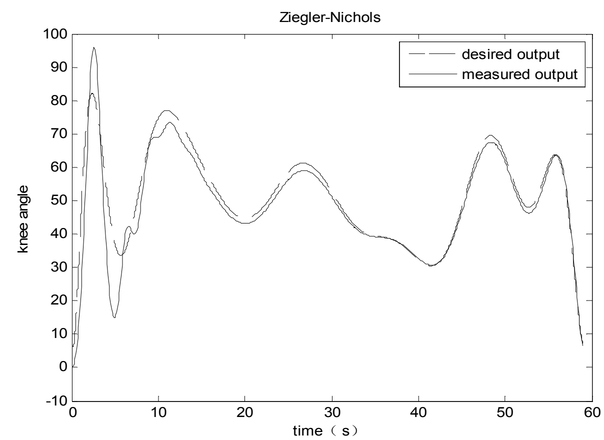

Fig.3. Knee angle controlled by the Ziegler-Nichols based PID

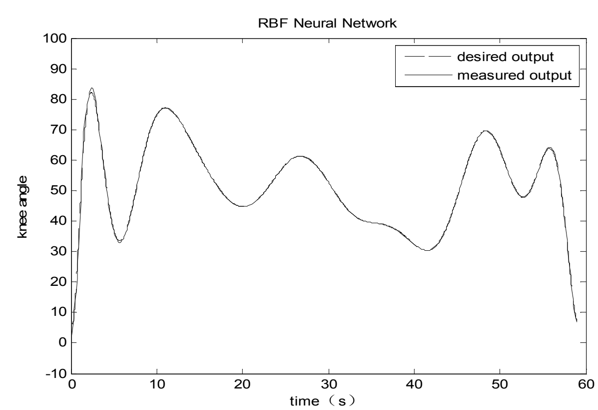

Fig.4. Knee angle controlled by the RBF based PID

In this experiment, Lilly wave, a balanced bidirectional pulse pair, with invariable pulse width (150μs) and amplitude ranged from 0 to 120 mA was used as a stimulation pulse. The stimulation frequency was 25 Hz and the knee joint angle sampling rate was 128 Hz. PID controllers based on RBF Neural Network as well as Ziegler-Nichols were applied to control the knee joint angle tracking the preset-trajectory respectively by modulating the pulse amplitude of the FES system.

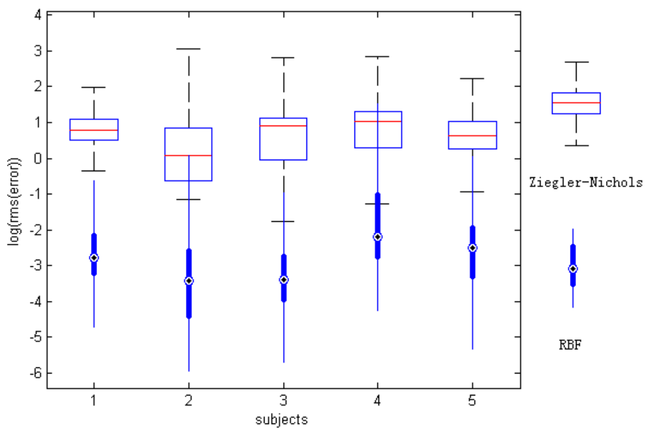

Fig.5 the errors controlled by both algorithms on the five subjects

Fig.3 shows the tracing result of the knee joint position controlled by Ziegler-Nichols PID controller on one subject among the five and Fig.4 shows that controlled by RBF Neural Network. The dashed line indicates the preset-trajectory and real line represents the measured output. We can see that from these figures the controllers tuning by RBF Neural Network overcomes the overshot which normally exists in the Ziegler-Nichols approach and may result in potential damage to stimulated muscle in FES system. The knee joint movement with RBF Neural Network can fit the preset-trajectory better than Ziegler-Nichols obviously. The errors controlled by both methods on the five subjects during the whole trails are shown in the Fig.5, and Root Mean Square and Nature Logarithm were imposed on for a significant discrimination. The rectangles boxes illuminate the results of the Ziegler- Nichols and the blue strips denote that of RBF Neural Network. The figure indicates that the latter one can maintain lower errors correspondingly and can improve the performance of PID to control the knee joint position dramatically with strong robustness.

IV. CONCLUSION

Conventional PID controller can hardly satisfied different operating condition [11]. This study has presented an adaptive FES-PID system that modulated by the RBF Neural Network to control knee joint position during quadriceps stimulation. The capability of the presented approach, with an advantage of self-learning, is superior to the traditional Ziegler- Nichols method to track the preset-trajectory, as indicated from the experimental results. In conclusion, an adaptive PID controller based on RBF Neural Network is an effective approach to improve the performance of the Functional Electrical Stimulation system. Moreover, it must be note that a suitable learning rate is important in practice, and we need to fix an optimum η according to different subjects in clinical application.

REFERENCES

[1] Gwo-Ching Chang, Jer-Junn Luh, Gon-Der Liao "A Neuro-Control System for the Knee Joint PositionControl with Quadriceps Stimulation" IEEE transactions on rehabilitation engineering, vol. 5, no. 1, march 1997 pp: 2-11

[2] J. M. Hausdorff and W. K. Durfee, "Open-loop position control of the knee joint using electrical stimulation of the quadriceps and hamstrings", Medical and Biological Engineering and Computing Volume 29, No.3, 1992 pp: 269-280

[3] McNeal, D.R. Nakai, R.J. and Meadows, "Open-loop control of the freely-swinging paralyzed leg" Biomedical Engineering, IEEE Transactions on. 1989 Volume: 36, Issue: 9, pp: 895-905

[4] Popovic, D.B. Stein, R.B. and Jovanovic, K.L. "Sensory nerve recording for closed-loop control to restore motor functions", Biomedical Engineering, IEEE Transactions on 1993, Volume: 40, Issue: 10pp: 1024-1031

[5] J. J. Abbas, "Feedback control of coronal plane hip angle in paraplegic subjects using functional neuromuscular stimulation," IEEE Trans.Biomed. Eng., 1991, vol. 38, pp. 687–698,.

[6] Kenji Kurosawa, Ryoko Futami, Takashi Watanabe. "Joint Angle Control by FES Using a Feedback Error Learning Controller". IEEE transactions on neural systems and rehabilitation engineering, vol. 13, no. 3, september 2005. pp: 359-371

[7] Veltink P.H., Chizeck H.J. and Crago P.E., 1992, Nonlinear joint angle control for artificially stimulated muscle. IEEE Trans. Biomed. Eng., 39, 68-380.

[8] Visioli, A. ''Tuning of PID controllers with fuzzy logic". [C] Control Theory and Applications, IEE Proceedings. 2002,148(1): 1-8.

[9] Benaskeur, A.R. Desbiens, A. "Backstepping-based adaptive PID control" [J] Control Theory and Applications. 2002, 149(1): 54-59.

[10] Yasue Mitsukura, Toru Yamamoto and Masahiro Kaneda "A Design of Self-Tuning PID Controllers Using a Genetic Algorithm" [C] Proceedings of the American Control Conference. 1999:1361-1365.

[11] Ming-guang Zhang, Xing-gui Wang, and Man-qiangLiu. "Adaptive PID Control Based on RBF Neural Network Identification" , Proceedings of the 17th IEEE International Conference on Tools with Artificial Intelligence (ICTAI'05) 2005, pp:680-683