

# An Adaptive RBF Neural Network Control Strategy for Lower Limb Rehabilitation Robot

Feng Zhang, Pengfeng Li, Zengguang Hou, Xiaoliang Xie, Yixiong Chen,  
Qingling Li, and Min Tan

Laboratory of Complex Systems and Intelligence Science, Institute of Automation,  
the Chinese Academy of Sciences, Beijing 100190, China

[zfeng1203@yahoo.com.cn](mailto:zfeng1203@yahoo.com.cn), [pengfeng.li@ia.ac.cn](mailto:pengfeng.li@ia.ac.cn), [hou@compsys.ia.ac.cn](mailto:hou@compsys.ia.ac.cn),  
[xiaoliang.xie@ia.ac.cn](mailto:xiaoliang.xie@ia.ac.cn), [yixiong.chen@ia.ac.cn](mailto:yixiong.chen@ia.ac.cn), [doudouhit@163.com](mailto:doudouhit@163.com),  
[min.tan@ia.ac.cn](mailto:min.tan@ia.ac.cn)

**Abstract.** This paper proposed an adaptive control strategy based on RBF (radial basis function) neural network and PD Computed-Torque algorithm for precise tracking of a predefined trajectory. This control strategy can not only give a small tracking error, but also have a good robustness to the modeling errors of the robot dynamics equation and also to the system friction. With this control algorithm, the robot can work in assist-as-needed mode by detecting the human active joint torque. At last, a simulation result using matlab simulink is given to illustrate the effectiveness of our control strategy.

**Keywords:** Rehabilitation robot, RBF, active training, adaptive control.

## 1 Introduction

There has been more and more patients in the world who suffered from SCI or stroke and the treatment to these patients is a long way after surgery. Conventionally speaking, doctors help these patients do rehabilitation training by hand or by simple device such as automated vehicles. These traditional treatments are useful to the rehabilitation of patients to some extent. But more and more studies proved that the active participation of patients in training can do a better favor to the neuro-rehabilitation. Literature [1][2] also suggest that active training can improve the cortical reorganization. Clinic trials have also proved that repetitive movement exercise is helpful to neurologic rehabilitation [3][4], even though that the scientific basis for neuro-rehabilitation remains ill-defined. So most of the limb rehabilitation robots also focus on the various movement training. Exoskeleton robots are mostly often used for these purpose.

Rehabilitation robot is designed for active training and repetitive movement exercise of patients who suffered from paraplegia or hemiplegia, so that the basic function of a rehabilitation robot is that the robot can detect the active participation of patients and can automatically assist the patients as needed. There are many ways to detect the active participation of patients, for example, the

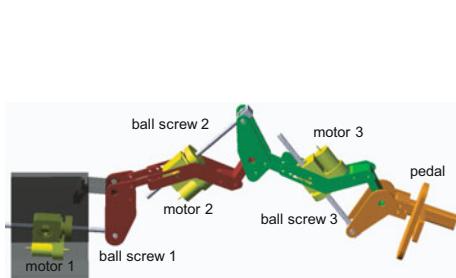
force sensor or torque sensor can be used to detect the patients' force that exerted onto the robot, the biological signal such as electromyography (EMG) or electroencephalogram (EEG) can be used to detect the intent of patients and also the status of muscles [5][6]. The MIT-MANUS which was designed by MIT in 1991 is an upper limb rehabilitation robot, it uses EEG to detect the patients' intent and hemiplegia patients can do some easy work with the help of the robot. The MotionMaker is an lower limb rehabilitation robot which was designed by SWORTEC, and EMG is used to detect the muscle activity. The Locomat is another lower limb rehabilitation robot which was designed by HOCOMA, it consists of a treadmill and a body weight support system. The Locomat robot can detect the torque the patients exerts onto the Locomat and then pull the patient's leg into the desired direction [7]. Most of the literatures about rehabilitation robots are mainly focus on the smart design or kinetic control [8, 9, 10], there is little literature about the dynamics control strategy to accomplish the precise trajectory tracking. In the following sections of this paper, we will give a introduction of our self-designed lower limb rehabilitation robot, and propose an adaptive control strategy based on RBF neural network and PD Computed-Torque algorithm for precise tracking of a predefined trajectory.

## 2 Simplified Dynamical Model of System

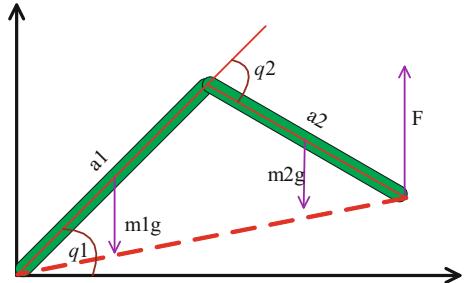
The self-designed 3-DOF exoskeleton lower limb rehabilitation robot model is showed in Fig.1. We only give an inner mechanical structure of the robot. The people's leg can be attached to the robot links by straps, and the foot can be fastened to the pedal. According to actual needs, the exoskeleton lower limb rehabilitation robot can help patients do various movement in a vertical plane, including single-joint movement and joint united movement. Three joints of the robot are all driven by ball screws. The ball screw is a line unit which changes the rotational motion of electrical motor into linear motion, and then the linear motion of the ball screw drives the link of the robot to rotate around the joint fulcrum. Compared with geared drive, this design can reduce the system friction, and increase the system damping, and reduce the influence of the external disturbance to the whole system. Three absolute encoders are installed in every joint to record the real joint angle, and the absolute encoders can maintain the current output even when the power is off. Also there is a force sensor installed at the end of each ball screw which is used to detect the joint torque. There is another more important force sensor placed on top of the foot treadle which is also the third link of the robot. This force sensor is used to detect the force the human foot exerts onto the foot treadle.

When using the platform for rehabilitation training, the length of each link should be adjusted firstly to set the link length equal to the human thigh and crus. Then the foot of the patient is fastened to the foot treadle. We can see the robot combined with the human leg as a whole, a three-link system. For simplification, the mass of human leg and the robot link are seen as evenly distributed, and the third link of the system is omitted because it just impacts

the foot posture and has no effect to the end trajectory. After simplification, the whole system can be seen as a well-proportioned two-link system showed in Fig. 2.



**Fig. 1.** 3-DOF rehabilitation robot for lower limbs



**Fig. 2.** Simplified model of system

## 2.1 Lagrange's Dynamics of Human Leg and Robot

By using Lagrange-Euler method, we can easily get the dynamics of the whole system, as showed in Fig. 2. Firstly, the Lagrange's equation of motion for the two-link system is defined as followed

$$\tau_i = \frac{d}{dt} \left[ \frac{\partial L}{\partial \dot{q}_i} \right] - \frac{\partial L}{\partial q_i} \quad (1)$$

where  $\tau_i$  is joint torque,  $q_i$  is the joint angle,  $L$  is the Lagrangian which represents the difference between the kinetic and potential energies

$$L = K - P \quad (2)$$

For link 1, the kinetic and potential energies can be given by

$$\begin{cases} K_1 = \frac{1}{6} m_1 a_1^2 \dot{q}_1^2 \\ P_1 = \frac{1}{2} m_1 g a_1 \sin q_1 \end{cases} \quad (3)$$

where  $m_1$  is the total mass of robot link 1 and human thigh,  $a_1$  is the length of link 1 and human thigh,  $q_1$  is the joint angle of hip. For link 2, the kinetic and potential energies can be given by

$$\begin{cases} K_2 = \left( \frac{1}{2} m_2 a_1^2 + \frac{1}{6} m_2 a_2^2 + \frac{1}{2} m_2 a_1 a_2 \cos q_2 \right) \dot{q}_2^2 \\ \quad + \left( \frac{1}{3} m_2 a_2^2 + \frac{1}{2} m_2 a_1 a_2 \cos q_2 \right) \dot{q}_1 \dot{q}_2 + \frac{1}{6} m_2 a_2^2 \dot{q}_1^2 \\ P_2 = m_2 g a_1 \sin q_1 + \frac{1}{2} m_2 g a_2 \sin(q_1 + q_2) \end{cases} \quad (4)$$

where  $m_2$  is the total mass of robot link 2 and human crus,  $a_2$  is the length of link 2 and human thigh,  $q_2$  is the joint angle of knee.

By combining (3) and (4), and computing the formula (1), we can get the ideal dynamics of the whole system which can be written in the standard form

$$M_0(q)\ddot{q} + V_0(q, \dot{q})\dot{q} + G_0(q) = \tau \quad (5)$$

with  $M_0(q)$  the inertia matrix,  $V(q, \dot{q})$  the Coriolis/centripetal vector, and  $G_0(q)$  the gravity vector. The specific form of these matrices are given in the simulation section. Note that  $M_0(q)$  is symmetric and positive definite, and  $\dot{M}(q) - V_0(q, \dot{q})$  is a skew-symmetric matrix, and the gravity vector  $G_0(q)$  is bounded [11].

## 2.2 Detect the Joint Torque of Human Leg

Actually, in formula (5), the torque  $\tau$  is comprised of two parts, one is caused by the voluntary movement of patient and the other is caused by robot motor. To detect the torque of human leg that exerts to the joint is very important and it will decide the output torque of robot motor.

We suppose that the human foot and the third link is always horizontal in training, so that the force of foot is always perpendicular to the ground, and the drawing force and compression force transducer on top of the third link can detect the the human voluntary force. From Fig.2, we can see that if there is no voluntary force, the reading of the transducer should be produced by the leg gravity. The relationship between them is

$$F_s = \frac{\left(\frac{1}{2}m_t + m_c\right)ga_1 \cos q_1 + \frac{1}{2}m_c g a_2 \cos(q_1 + q_2)}{a_1 \cos q_1 + a_2 \cos(q_1 + q_2)} \quad (6)$$

With  $F_s$  the reading of transducer at static status,  $m_t$  and  $m_c$  the mass of thigh and crus. If the patient exerts voluntary force to the pedal, the reading of the transducer should be bigger or smaller than  $F_s$ . From the reading of transducer we can judge the system running mode. Specifically, if the reading is between 0 and  $F_s$ , then the robot arm will work in the assistance mode, else if the reading is bigger than  $F_s$  or less than 0, then the robot arm will work in the resistance mode. If the current reading of transducer is  $F$ , then the difference between  $F_s$  and  $F$  is the patient's voluntary force. Note that the leg gravity is not voluntary force. By transforming the force into joint torque, we can get

$$\tau_h = \begin{bmatrix} (F_s - F)a_1 \sin(q_1 + q_2) \sin q_2 \\ (F_s - F)a_2 \cos(q_1 + q_2) \end{bmatrix} \quad (7)$$

where  $\tau_h$  is patient's voluntary torque.

## 2.3 Control Strategy

In practice, the precise model of the plant is hard to get, and we can only build the ideal model. If the model (5) we have built is precise, then a Computed-PD

algorithm as followed is enough to make a good tracing of the predefined trajectory [11].

$$\tau = M_0(q)(\ddot{q}_d - K_v \dot{e} - K_p e) + V_0(q, \dot{q})\dot{s} + G_0(q) \quad (8)$$

where  $q_d$  is the desired trajectory, and  $e = q - q_d$ ,  $\dot{e} = \dot{q} - \dot{q}_d$ , the derivative gain matrix  $K_v$  and the proportional gain matrix  $K_p$  are selected positive definite.

But if the true model is

$$M(q)\ddot{q} + V(q, \dot{q})\dot{q} + G(q) + F(q, \dot{q}) = \tau \quad (9)$$

where  $F(q, \dot{q})$  is friction and disturbance item. Put the control law (8) into (9), and we get

$$\ddot{e} + K_v \dot{e} + K_p e = M_0^{-1}(\Delta M \ddot{q} + \Delta V \dot{q} + \Delta G - F(q, \dot{q})) \quad (10)$$

where  $\Delta M = M_0 - M$ ,  $\Delta V = V_0 - V$ , and  $\Delta G = G_0 - G$ . Easily we can see that the control law (8) is not satisfying. The unprecise part of the model is the right item of (10). We define the unprecise part of the model as

$$f(x) = M_0^{-1}(\Delta M \ddot{q} + \Delta V \dot{q} + \Delta G - F(q, \dot{q})) \quad (11)$$

If we have known the unprecise part, then we can design the controller as

$$\tau = M_0(q)(\ddot{q}_d - K_v \dot{e} - K_p e - f(x)) + V_0(q, \dot{q})\dot{s} + G_0(q) \quad (12)$$

Substitute (12) into (9), we can get the following error differential equation

$$\ddot{e} + K_v \dot{e} + K_p e = 0 \quad (13)$$

From this error differential equation, we can see that if the derivative gain matrix  $K_v$  and the proportional gain matrix  $K_p$  are selected positive definite, the system must converge within a certain time.

Unfortunately, the unprecise part of model is not available more often. The common method to resolve this problem is to compensate for the unprecise part of model. Neural network is a common method to approximate the unprecise item of a system [11]. The BP network and RBF network are two more often used neural networks. The BP network is a global approximation network, and the convergence is slow and be easy to fall into local minimum. On the contrary, the RBF network is a local approximation network, and has a fast learning speed, also it can avoid the local minimum [12]. Here we choose RBF neural network to approximate the unprecise part of the model. The structure of RBF neural network is showed in Fig.3. The gauss basis function is also chosen as followed

$$\phi_j = \exp\left(-\frac{\|X - c_j\|^2}{2b_j^2}\right), j = 1, 2, \dots, m \quad (14)$$

In the RBF network,  $X = [x_1, x_2, \dots, x_n]^T$  is the input vector,  $c_j$  is the central vector of node  $j$ ,  $c_j = [c_{j1}, c_{j2}, \dots, c_{jn}]$ , and  $b_j$  is the base width of node  $j$ .

$\varphi = [\phi_1, \dots, \phi_m]^T$  is the radial basis vector. If the weights vector of network is  $W = [w_1, \dots, w_m]^T$ , then the output of network is

$$y(t) = w_1\phi_1 + w_2\phi_2 + \dots + w_m\phi_m \quad (15)$$

In our control system, we choose  $X = [e \dot{e}]^T$  as input vector,  $\hat{W}$  as the weights vector which will be adjusted online,  $W^*$  is the best approximation weight vector, and the network output is  $\hat{f}(x, w) = \hat{w}^T \varphi(x)$ , and  $\hat{W}$  is the estimation value of  $W^*$ . At the basis of (8–10), we design the controller as

$$\tau = M_0(q)(\ddot{q}_d - K_v \dot{e} - K_p e - \hat{f}(x, w)) + V_0(q, \dot{q})\dot{s} + G_0(q) \quad (16)$$

If we substitute (16) into (9), then we will get

$$\dot{x} = Ax + B\{f(x) - \hat{f}(x, w)\} \quad (17)$$

where  $f(x) = M_0^{-1}(\Delta M \ddot{q} + \Delta V \dot{q} + \Delta G - F(q, \dot{q}))$ , and  $A = \begin{bmatrix} 0 & I \\ -k_p & -k_v \end{bmatrix}$ ,  $B = \begin{bmatrix} 0 \\ I \end{bmatrix}$ .

If we define the Lyapunov function as

$$V = \frac{1}{2}x^T Px + \frac{1}{2\gamma}\|\tilde{w}\|^2 \quad (18)$$

where  $P$  is a positive definite symmetric matrix, and it satisfies the Lyapunov function as followed

$$PA + A^T P = -Q \quad (19)$$

where  $Q \geq 0$ . If we choose the adaptive law as

$$\dot{\tilde{w}} = \gamma\varphi(x)x^T PB + k_1\gamma\|x\|\hat{w} \quad (20)$$

with  $\gamma > 0, k_1 > 0$ . The condition of convergence is

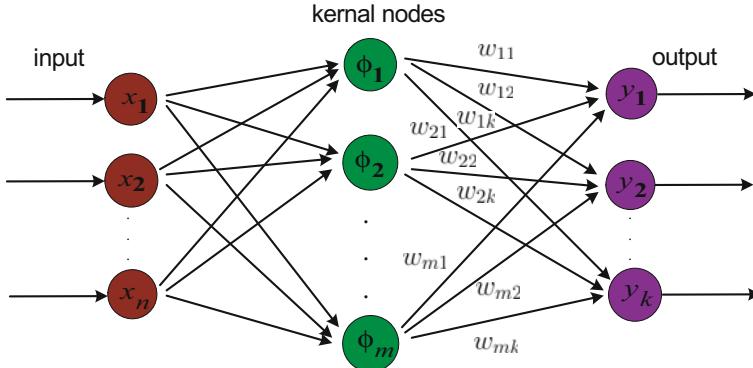
$$\|x\| \geq \frac{2}{\lambda_{min}(Q)} \left( \|\eta_0\| \lambda_{max}(P) + \frac{k_1}{4} w_{max}^2 \right) \quad (21)$$

with  $\eta_0$  the ideal error between  $f(x)$  and  $\hat{f}(x, w^*)$ .

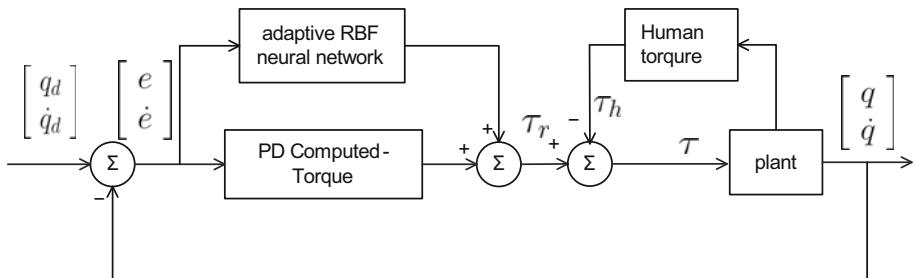
With above discussions, we can get the control system block diagram as Fig.4.

### 3 Simulation

In order to verify the resulting performance and robustness of the control algorithm, the simulation is conducted using SIMULINK toolboxes of MATLAB. The namely model we use is the model we have built as (8), where the inertia matrix  $M_0(q)$  is given by  $M_0(q) = \begin{bmatrix} \alpha + 2\eta \cos q_2 & \beta + \eta \cos q_2 \\ \beta + \eta \cos q_2 & \beta \end{bmatrix}$ . The centrifugal/Coriolis torque  $V_0(q, \dot{q})$  is given by  $V_0(q, \dot{q}) = \begin{bmatrix} -\eta \sin q_2 \dot{q}_2 & -\eta \sin q_2 \dot{q}_1 - \eta \sin q_2 \dot{q}_2 \\ \eta \sin q_2 \dot{q}_1 & 0 \end{bmatrix}$ . The



**Fig. 3.** The radial-basis-function network



**Fig. 4.** Control system block diagram

gravity  $G_0(q)$  is given by  $\begin{bmatrix} (\frac{1}{2}m_1 + m_2)ga_1 \cos q_1 + \frac{1}{2}m_2ga_2 \cos(q_1 + q_2) \\ \frac{1}{2}m_2ga_2 \cos(q_1 + q_2) \end{bmatrix}$ . The real model we use is (9), with  $M(q)$ ,  $V(q, \dot{q})$ ,  $G(q)$  are all twenty percent less than that of  $M_0(q)$ ,  $V_0(q, \dot{q})$ ,  $G_0(q)$  for simulation, the disturbance and friction item is chosen as

$$F(q, \dot{q}) = k_r \dot{q} \quad (22)$$

The following parameters are used in the simulation:  $m_1 = 15.2\text{kg}$ ,  $m_2 = 12.51\text{kg}$ ,  $a_1 = 0.42\text{m}$ ,  $a_2 = 0.41\text{m}$ . The parameters  $\alpha$ ,  $\beta$  and  $\eta$  in  $M_0(q)$  and  $V_0(q, \dot{q})$  are representative of  $\alpha = (\frac{1}{3}m_1 + m_2)a_1^2 + \frac{1}{3}m_2a_2^2$ ,  $\beta = \frac{1}{3}m_2a_2^2$  and  $\eta = \frac{1}{2}m_2a_1a_2$ . The coefficient associated with the friction is  $k_r = 1.2$ .

Treadmill exercise is a mostly often used movement to paraplegia or hemiplegia patients, and it has been proven very effective to their recovery. In our simulation, we also choose this movement style which means that the trajectory of the rehabilitation robot end is a circle. Specifically speaking, we choose the trajectory as

$$\begin{cases} x = 0.63 + 0.1 \cos(0.5\pi t) \\ y = 0.63 + 0.1 \sin(0.5\pi t) \end{cases}$$

The center of the circle is selected as  $(0.63, 0)$ , and the radius is  $0.1m$ . According to [13], we know that the human walking speed ranges from  $67 \pm 3$  (very slow) to  $154 \pm 11$  steps/min (very fast), and a normal person walks  $100 \pm 1$  steps/min at the usual walking speed. We set the simulation period as  $4s$ , and it is suitable for a injured person. From the end trajectory of the robot, we can get the desired trajectory of every joint. It can be written as followed:

$$\begin{cases} q_2^d = -\arccos \frac{x^2 + y^2 - a_1^2 - a_2^2}{2a_1a_2} \\ q_1^d = \arcsin \frac{y}{\sqrt{x^2 + y^2}} - \arctan \frac{a_2 \sin(q_2^d)}{a_1 + a_2 \cos(q_2^d)} \end{cases}$$

The  $q_1^d$  is the desired trajectory of joint 1, and  $q_2^d$  is the desired trajectory of joint 2.

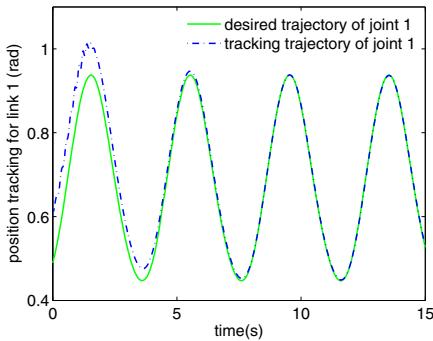
The torque that human exert onto the pedal is chosen as 20% of the system output torque for simulation. The gain matrices  $K_p$  and  $K_v$  are selected diagonal as  $K_p = \begin{bmatrix} 16 & 0 \\ 0 & 16 \end{bmatrix}$ ,  $K_v = \begin{bmatrix} 8 & 0 \\ 0 & 8 \end{bmatrix}$  respectively. The number neurons of the RBF

hidden layer we choose is four and the central vector is  $c = \begin{bmatrix} -4 & -2 & 0 & 2 \\ -4 & -2 & 0 & 2 \\ -4 & -2 & 0 & 2 \\ -4 & -2 & 0 & 2 \end{bmatrix}$ , the base width of neurons is  $b_j = [5 \ 5 \ 5 \ 5]$ . The parameters in (20) are chosen as

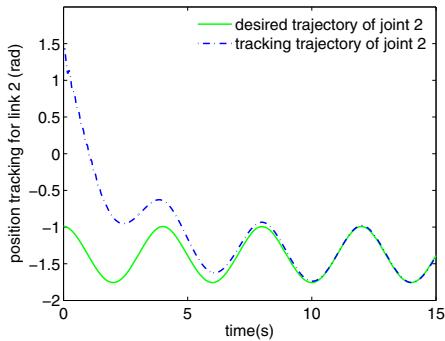
$\gamma = 100$ ,  $k_1 = 0.001$ , and the Lyapunov matrix as  $Q = \begin{bmatrix} 60 & 0 & 0 & 0 \\ 0 & 60 & 0 & 0 \\ 0 & 0 & 60 & 0 \\ 0 & 0 & 0 & 60 \end{bmatrix}$ . The

parameters above have different influences to the simulation. Specifically, the much bigger of the gain matrices  $K_p$  and  $K_v$  the faster convergence of trajectory. The  $\gamma$  and Lyapunov matrix  $Q$  is much bigger and the  $k_1$  is much smaller then the compensation for the unprecise part is more precise. Besides, the central vector and base width of neurons have small influence to the simulation. These conclusions are consistent with formula (13) and (21).

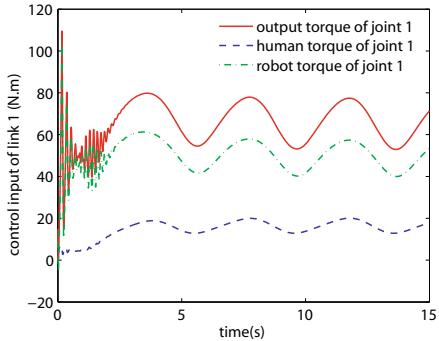
Figs 5 and 6 depict the position tracking results in 15 sec. The tracking error is almost less than 0.005 rad (0.287 deg) after 9 sec for both joints. Figs 7 and 8 show the output torque for tracking the desired trajectory. It can be seen that the output torque is the combination of human torque and robot torque. From figs 7 and 8 we also can see that the robot is working in assistance mode for the reason that the human torque and the robot torque are of the same direction. Once the human torque and the robot torque are of different directions, then the robot will work in resistive mode. Thus the assist-as-needed mode is accomplished. While in rehabilitation training for patients who have suffered from different degrees of disability, once the doctors have set the trajectory and the speed, the robot will decide to help or resist the patient according to the patients' degree of voluntary. If the patient can not control his lower limbs to follow the trajectory, then the robot will give him a hand. On the contrary, if the patient can not only



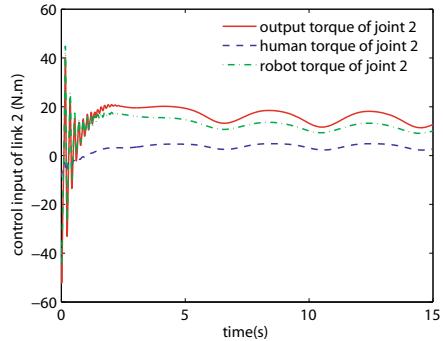
**Fig. 5.** Desired and actual trajectory of joint 1



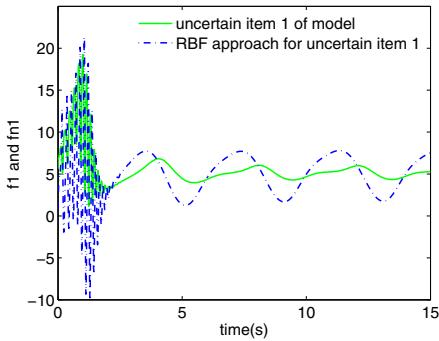
**Fig. 6.** Desired and actual trajectory of joint 2



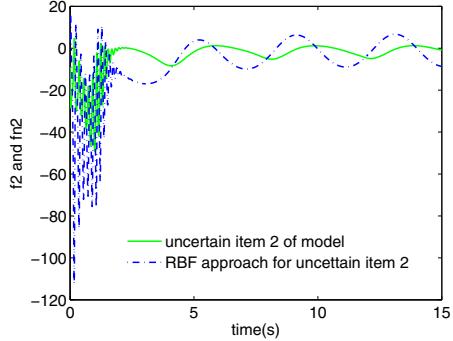
**Fig. 7.** Output torque of joint 1



**Fig. 8.** Output torque of joint 2



**Fig. 9.** RBF neural network approximation for uncertain item of joint 1



**Fig. 10.** RBF neural network approximation for uncertain item of joint 2

follow the trajectory but also want to move faster, then the robot will give him an resistance. So the whole system will always work in the predefined form the doctors have set. Figs 9 and 10 show the approximation to the unprecise part of the dynamics model as showed in formula (10).

## 4 Conclusion

In this paper, an adaptive RBF neural network combined with PD Computed-torque control algorithm is proposed for trajectory tracking of lower limbs rehabilitation robot. In this control algorithm, the RBF neural network is used to approximate the uncertain part of the dynamics model, including model error, friction and external disturbance, and the PD Computed-torque is used to control the known part of dynamics. From the simulation result we can see that the RBF approximation has a good performance and the control algorithm has a good robustness to the load changes and system friction. As is known to all that the rehabilitation robot load is different for different patients, and the RBF compensation can resolve the influence of physical difference. Furthermore, using this control algorithm, the robot can work in the assist-as-needed mode by detecting the joint torque. The control algorithm is just for simulation and has not been used in practice to date.

## Acknowledgements

This research is supported in part by the National Natural Science Foundation of China (Grant #60775043), the National Hi-Tech R & D Program (863) of China (Grant #2009AA04Z201), and the Sci. & Tech. for the Disabled Program of the Chinese Academy of Sciences (Grant #KGCX2-YW-618).

## References

- [1] Lotze, M., Braun, C., Birbaumer, N., Anders, S., Cohen, L.G.: Motor learning elicited by voluntary. *Brain* 126, 866–872 (2003)
- [2] Schoone, M., van Os, P., Campagne, A.: Robot-mediated Active Rehabilitation (ACRE) A user trial. In: IEEE 10th international conference on rehabilitation robotics, pp. 477–481 (2007)
- [3] Pohl, M., Werner, C., Holzgraefe, M., Kroczeck, G., Mehrholz, J., Wingendorf, I., Holig, G., Koch, R., Hesse, S.: Repetitive locomotor training and physiotherapy improve walking and basic activities of daily living after stroke: a single-blind, randomized multicentretrial (DEutsche GAngtraineRStudie, DEGAS). *Clinical Rehabilitation* 21, 17–27 (2007)
- [4] Hasan, M.K., Park, S.-H., Seo, S.-J., Sohn, D.-H., Hwang, S.-H., Khang, G.: A Gait Rehabilitation and Training System based on Task Specific Repetitive Approach. In: IEEE 3rd international conference on bioinformatics and biomedical engineering, pp. 1–4 (2009)

- [5] Yang, Q., Siemionow, V., Yao, W., Sahgal, V., Yue, G.H.: Single-trial EEG-EMG coherence analysis reveals muscle fatigue-related progressive alterations in corticomuscular coupling. *IEEE transactions on neural systems and rehabilitation engineering* 18(2), 97–106 (2010)
- [6] Hincapie, J.G., Kirsch, R.F.: Feasibility of EMG based neural network controller for an upper extremity neuroprosthesis. *IEEE transactions on neural systems and rehabilitation engineering* 17(1), 80–90 (2009)
- [7] Lunenburger, L., Colombo, G., Riener, R., Dietz, V.: Biofeedback in gait training with the robotic orthosis locomat. In: The 26th Annual International conference of the IEEE EMBS, vol. 7, pp. 4888–4891 (2004)
- [8] Den, A., Moughamir, S., Afilal, L., Zaytoon, J.: Control system design of a 3-dof upper limbs rehabilitation. *Computer methods and programs in biomedicine* 89, 202–214 (2007)
- [9] Jezernik, S., Scharer, R., Colombo, G., Morari, M.: Adaptive robotic rehabilitation of locomotion: a clinical study in spinally injured individuals. *Spinal cord* 41, 657–666 (2003)
- [10] Morita, Y., Hirose, A., Uno, T., Uchida, M., Ukai, H., Matsui, N.: Development of rehabilitation training support system using 3D force display robot. Springer, Heidelberg (2007)
- [11] Lewis, F.L., Dawson, D.M., Abdallah, C.T.: Robot manipulator control theory and practice. Marcel Dekker, Inc., New York (2004)
- [12] Park, J., Sandberg, I.W.: Universal approximation using radial-basis-function networks. *Neural Computation* 3, 246–257 (1991)
- [13] Mark, D.L., Hylton, B.M., Victor, S.F., Stephen, R.L.: Walking speed, cadence and step length are selected to optimize the stability of head and pelvis accelerations. *Experimental Brain Research* 184(2), 201–209 (2008)