A Particle Swarm Optimized Fuzzy Neural Network Control for Acrobot

Dong-bin Zhao and Jian-qiang Yi

Laboratory of Complex Systems and Intelligence Science, Institute of Automation,
Chinese Academy of Sciences, Beijing 100080, P.R. China
{Dongbin.zhao, jianqiang.yi}@mail.ia.ac.cn

Abstract. This paper addresses the problem of controlling an acrobot, an under-actuated robotic systems, using fuzzy neural network approach. A five-layer Takagi-Sugeno fuzzy neural network control (TSFNNC) is proposed to swing up the acrobot from the low stable equilibrium to approach and balance around its top unstable equilibrium position. By analyzing the system dynamics, total energy and potential energy of the system are introduced in the second layer, with the system states as the inputs to the first layer. Fuzzy membership functions and rules are depicted in the third and fourth layers respectively. The fifth layer works as the final output. A modified particle swarm optimizer (PSO) is adopted to train the consequents in the fourth layer. Simulation results indicate that the integrated TSFNNC approach can control the acrobot system from upswing to balance process effectively. This approach provides an easy and feasible solution for similar control problems.

1 Introduction

Acrobot (**Acro**batic Ro**bot**) is a typical structure of two link under-actuated robotic systems, which are under extensive investigations recently [1-7]. The upswing control of the acrobot is to rotate the two links from the low stable to the top unstable equilibrium position. No smooth controller is proved to realize the whole process. A common adopted control scheme is to swing up the system close to the top unstable equilibrium with a nonlinear controller, and then switch to the other balance controller.

Spong [1] proposed the concept of partial feedback linearization, to drive the two links into the attractive basin of a linear balance controller. Smith, et al. [2] presented a dynamic fuzzy controller integrated with genetic algorithm and dynamic switching systems. Lai, et al. [3] developed a model-free controller for upswing and model-based controller for balancing. Xu, et al. [4] designed a time-optimal controller based on neural network reinforcement learning to reduce upswing time. Aiming at real application, Zhao and Yi [5] proposed a GA tuned fuzzy neural network control approach with limited torque output.

But during the switching process from upswing to balance, the control torque is usually not very smooth. Therefore, a further investigation concentrates on an integrated control approach for the whole process without control switching. Wang etc. [6] provided an adaptive sliding mode control. This paper is to develop a new integrated control approach based on computational intelligence. A five-layer Takagi-Sugeno fuzzy neural network control (TSFNNC) is proposed. Total energy and

J. Wang et al. (Eds.): ISNN 2006, LNCS 3972, pp. 1160 – 1165, 2006.

potential energy of the system are introduced in the second layer, with the system states as the inputs to the first layer. Fuzzy membership functions and rules are depicted in the third and fourth layers respectively. The fifth layer works as the final output. A modified particle swarm optimization (PSO) is adopted to train the consequents in the fourth layer.

2 System Dynamics

The model of the acrobot is shown in Fig. 1.

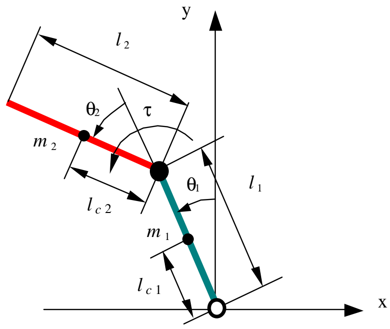

Fig. 1. Model of the acrobot system

Supposing there is no friction, the system dynamics equation is expressed by

$$\boldsymbol{\tau} = D(\boldsymbol{\theta})\boldsymbol{\alpha} + C(\boldsymbol{\theta},\boldsymbol{\omega})\boldsymbol{\omega} + G(\boldsymbol{\theta}) \tag{1}$$

Where $\boldsymbol{\tau}$ is the external torque, which is only applied on the second joint. The variables $\boldsymbol{\theta}=[\theta_1\ \theta_2]^T$, $\boldsymbol{\omega}=[\omega_1\ \omega_2]^T$, and $\boldsymbol{\alpha}=[\alpha_1\ \alpha_2]^T$ are angles, angular velocities and angular accelerations of the two links respectively. The terms D[$\boldsymbol{\theta}$], C[$\boldsymbol{\theta}, \boldsymbol{\omega}$] and G[$\boldsymbol{\theta}$] can be represented with five variables $\{q_1, q_2, q_3, q_4, q_5\}$. Detailed description of these parameters can be also seen in [5].

The kinetic energy E_k, potential energy E_p, and the total energy E are depicted by

$$\begin{aligned} E_k &= \frac{1}{2}\omega^T D(\theta)\omega \\ E_p &= q_4 g \cos\theta_1 + q_5 g \cos(\theta_1 + \theta_2) \\ E &= E_k + E_p \end{aligned} \tag{2}$$

One property of the two link under-actuated system is the passivity [7], which can be derived from equations 1 and 2, depicted as follows.

$$\dot{E} = \tau_2\, \omega_2 \tag{3}$$

It means that system energy will increase when the torque is exerted on the active joint in the same direction with its angular velocity, vice versa.

State vector **X** is selected as $[\theta_1,\ \omega_1,\ \theta_2,\ \omega_2]^T$. θ_1 and θ_2 vary within [-π , π]. From equ.2, four equilibrium positions of the system can be derived, of which the commonly reachable two are as follows.

(0, 0, 0, 0): Top unstable position with the top energy E_{top}, $E = E_{top} = q_4 g + q_5 g$

(π, 0, 0, 0): Low stable position with the low energy, $E = -E_{top} = -(q_4 g + q_5 g)$

The top unstable equilibrium position is the most difficult case for feedback stabilization among all the equilibriums. The integrated upswing control is to move the two links from the low stable to the top unstable position, and maintain there.

3 Particle Swarm Optimized Fuzzy Neural Network Control

From the analysis of the system dynamics, we can see that it will be a feasible way to increase the system energy when the energy is low by apply appropriate torque on the active joint. It is desirable to increase the total energy and potential energy simultaneously. When they both reach the top energy E_{top}, which is defined as an attractive region of linear controller [6], the system will be balanced easily.

To realize the above control scheme, a five-layer Takagi-Sugeno Fuzzy Neural Network is proposed, whose architecture is shown in Fig. 2. Each layer outputs serve as the inputs of its next layer. Layers functions are given as follows.

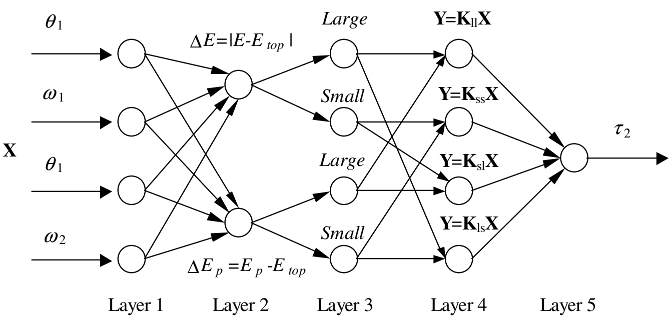

Fig. 2. TSFNNC Structure

Layer 1: Inputs

The state space variables θ_1, ω_1, θ_2, and ω_2 are as the inputs.

Layer 2: Energy error calculation

The absolute errors ΔE and ΔE_p are calculated. When they are small enough to zeros, the system approaches the top unstable position. When the errors are both large to E_{top}, the system is at the initial low stable position. When one of the errors is small and the other is large, they correspond to different system states, shown in Fig.3.

Layer 3: Membership definition

Two fuzzy subsets *Large* and *Small* are defined for each energy error. Triangle functions are adopted to describe the membership degree, shown in Fig. 4. The scope of

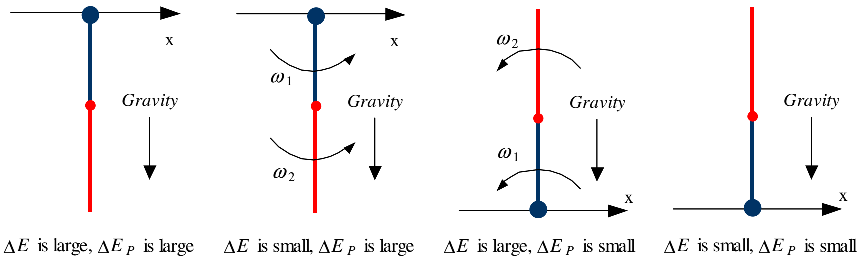

Fig. 3. Different state spaces of the acrobot system depicted by energy error

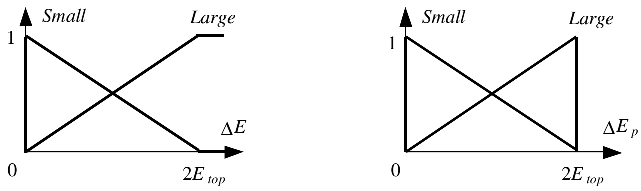

Fig. 4. Membership functions

the potential energy error ΔE_p is [0, $2E_{top}$]. But the total energy error ΔE may be over $2E_{top}$, where the degree of the membership *Large* and *Small* is set to 1 and 0 respectively. For each energy error, the sum of the degrees is 1.

Layer 4: Fuzzy rules

There are only four fuzzy rules, corresponding to different sates.

$$\begin{aligned}
&\text{Rule 1: If } \Delta E \text{ is large and } \Delta E_p \text{ is large, then } Y_1 = \mathbf{K_{ll}X} = sign(x(4))\tau_{\max} \\
&\text{Rule 2: If } \Delta E \text{ is small and } \Delta E_p \text{ is large, then } Y_2 = \mathbf{K_{ls}X} \\
&\text{Rule 3: If } \Delta E \text{ is large and } \Delta E_p \text{ is small, then } Y_3 = \mathbf{K_{sl}X} = -sign(x(4))\tau_{\max} \\
&\text{Rule 4: If } \Delta E \text{ is small and } \Delta E_p \text{ is small, then } Y_4 = \mathbf{K_{ss}X}
\end{aligned} \tag{4}$$

where Rule 1 depicts the action on the initial low stable position, so the consequent is determined to the maximum output torque with the same direction as the angular velocity to increase the system energy. On the other side, Rule 3 depicts the action on the system with high energy, so it is just opposite to Rule 1. Rule 4 describes the system around the top unstable position, so the consequent can be derived with linear feedback controller. Rule 2 describes the system in upswing process, so only the feedback parameters of this consequent need to be determined. The optimum consequent will help to swing up the system quickly and smoothly.

Layer 5: Output

Due to the definition of the fuzzy membership, the final output becomes the summation of the outputs of layer 4. The output is saturated to the positive or negative maximum torque if it is beyond the torque limit.

Particle swarm optimization, proposed recently, shows good performance for fast speed and low error in neural network training and other application fields. Aiming at the consequents of the fuzzy neural network control to be determined, a modified particle swarm optimization algorithm [8] is adopted.

$$v_{id} = \omega v_{id} + c_1 r_1 \left(p_{id} - x_{id}\right) + c_2 r_2 \left(p_{gd} - x_{id}\right),\ x_{id} = x_{id} + v_{id} \tag{3}$$

where x_{id} is the current position of particle i, to represent the consequent $\mathbf{K}_{\mathbf{ls}}$. v_{id} is the velocity of the particle i. p_{id} is the best position of particle i, and p_{gd} is the best position of the whole particle swarm, to represent the final problem solution. c_1 and c_2 are learning factors. r_1 and r_2 are two random number. w is a velocity weight.

To evaluate the particle performance, the fitness is defined by

$$fitness = \exp\left(-\sqrt{\sum_{t=1}^{N} e(t)}\right),\ e(t) = \Delta E + \Delta E_p \tag{4}$$

where the total calculation step N is set to a large number, to provide enough time to swing up the system.

4 Simulation Results

The acrobot structure parameters are selected the same as [1] for simulation comparison. The coefficients $\{q_1, q_2, q_3, q_4, q_5\}$ are $\{1.3333, 1.3333, 1, 1.5, 1\}$. The maximum torque $\tau_{\max}$ is 4 Nm, and $\mathbf{K}_{\mathbf{ss}}$ is determined by LQR mechanism.

$$\mathbf{K}_{\mathbf{ss}} = \left[-245.9821, -106.2845, -98.4906, -50.0736\right] \tag{5}$$

The scope of the consequent $\mathbf{K}_{\mathbf{ls}}$ is [-10,10], among which the best is found by the modified particle swarm optimization. 50 particles are initialized, and 100 generations are calculated. The total calculation step N is 1000, and sampling time is 0.01s. The velocity weight w decreases from 0.9 to 0.4 with the generation increasing. The learning factors c_1 and c_2 are both 2.0. The maximum v_{id} is limited within [-2,2]. Several experiments all derive feasible $\mathbf{K}_{\mathbf{ls}}$ solutions, one of which is selected as

$$\mathbf{K}_{\mathbf{ls}} = \left[-1.6785, -2.0542, -1.8864, -1.9806\right] \tag{6}$$

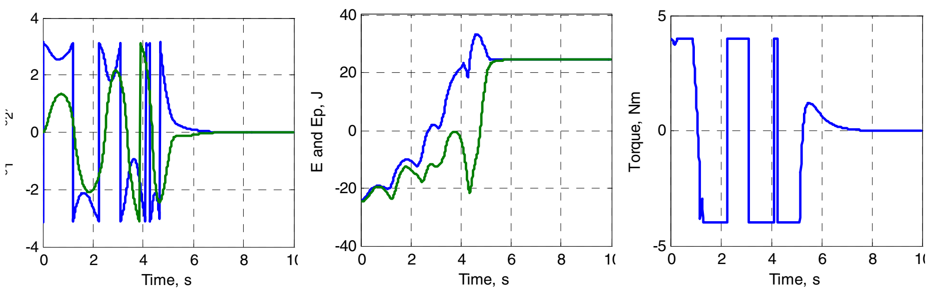

Fig. 5. Simulation results to swing up the acrobot. (*a*) θ_1 and θ_2; (*b*) E and E_p; (*c*) τ_2

Simulation results are shown in Fig. 5. It can be seen that system is swung up to its top unstable equilibrium position successfully. The torque is quite smaller compared to [1], which requires up to hundreds Nm. The torque also varies more smoothly compared to switch control method [5]. During several swings, the system energy E is pumped up effectively to E_{top}, which characterizes the method as an energy approach.

5 Conclusions

An integrated five-layer Takagi-Sugeno fuzzy neural network control is proposed. The consequent is learned with a modified particle swarm optimization algorithm. An acrobot is effectively swung up from the low stable equilibrium to approach and balance around its top unstable equilibrium with a limit torque for a short time. The control scheme can also be extended to other under-actuated systems. The control stability and other modified particle swarm optimizations will be further investigated in future research.

Acknowledgement

This work is supported by NSFC Projects (No.60475030 and No.60575047), and MOST Projects (No. 2003CB517106 and 2004DFB02100).

References

1. Spong, M.W.: The Swing up Control Problem for the Acrobot. IEEE Control Systems Magazine 15(1) (1995) 49-55
2. Smith, M.H., Zhang, T., Gruver, W.A.: Dynamic Fuzzy Control and System Stability for the Acrobot. IEEE International Conference on Fuzzy Systems (1998) 286-291
3. Lai, X., She, J.H. Ohyama, Y., Cai, Z.: Fuzzy Control Strategy for Acrobots Combining Model-free and Model-based Control. IEE Proceedings on Control Theory and Applications 146(6) (1999) 505-510
4. Xu, X., He, H.G.: Residual-gradient-based Neural Reinforcement Learning for the Optimal Control of an Acrobot. IEEE International Symposium on Intelligent Control (2002) 758-763
5. Zhao, D.B., Yi, J.Q.: GA-based Control to Swing up an Acrobot with Limited Torque. Proceedings of International Conference on Intelligent Computing (2005) 3358-3367
6. Wang, W., Yi, J.Q., Zhao, D.B., Liu, D.T.: Design of a New Stable Sliding-mode Controller for a Class of Second-order Underactuated Systems. IEE Proceedings – Control and Applications 151(6) (2004) 683-690
7. Fantoni, I., Lozano, R., Spong, M.W.: Energy Based Control of the Pendubot, IEEE Transactions on Automatic Control 45(4) (2000) 725-729
8. Shi, Y., Eberhart, R.: A Modified Particle Swarm Optimizer□IEEE World Congress on Computational Intelligence (1998) 69-73