

## Research Article

# Simple Model-Free Controller for the Stabilization of Planetary Inverted Pendulum

Huanhuan Mai,<sup>1</sup> Ying-Jeh Huang,<sup>2</sup> Xiaofeng Liao,<sup>1</sup> and Ping-Chou Wu<sup>2</sup>

<sup>1</sup>Department of Computer, Chongqing University, Chongqing 400044, China

<sup>2</sup>Department of Electrical Engineering, Yuan Ze University, Chung-Li 32003, Taiwan

Correspondence should be addressed to Huanhuan Mai, maihuan123@126.com

Received 2 May 2012; Revised 16 July 2012; Accepted 8 August 2012

Academic Editor: Wen Yu

Copyright © 2012 Huanhuan Mai et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

A simple model-free controller is presented for solving the nonlinear dynamic control problems. As an example of the problem, a planetary gear-type inverted pendulum (PIP) is discussed. To control the inherently unstable system which requires real-time control responses, the design of a smart and simple controller is made necessary. The model-free controller proposed includes a swing-up controller part and a stabilization controller part; neither controller has any information about the PIP. Since the input/output scaling parameters of the fuzzy controller are highly sensitive, we use genetic algorithm (GA) to obtain the optimal control parameters. The experimental results show the effectiveness and robustness of the present controller.

## 1. Introduction

To ensure better performance in diverse operating conditions, more and more modern control systems emerged. However, these control systems may fall outside of the scope of conventional control. Different from traditional model-based control, modern control exhibits less dependence on mathematical models. The model-free concept has been proposed that the controller does not contain any information about the system to be controlled [1]. Taking advantage of the concept, many high nonlinear systems are successfully to be controlled. Coelho et al. proposed a model-free learning adaptive control (MFLAC) strategy that is based on pseudogradient concepts with compensation using an RBF neural network and DE optimization [2]. An adaptive higher-order differential feedback controller which does not depend on the model of the controlled chaotic system has also been studied [3]. These controllers inherently exhibited its robustness.

In the field of intelligent control, fuzzy logic control (FLC) has gained popularity as a model-free approach that often outperforms other conventional approaches such as nonlinear adaptive control or PID controls [4]. FLC provides a framework for approximate reasoning and allows expert knowledge to be translated into an executable rule set. It

could deal with vague and incomplete information and exhibit robustness to noise and variations in system parameters [5]. Furthermore, when the system is too complex [6] or has high degree of nonlinearity [4] and the underlying processes are insufficiently understood [7], FLC plays an important role in robust control.

However, fuzzy systems have a well-known problem relating to the determination of their parameters: the membership functions, scaling factors, and rule base. To achieve better performance and improved robustness, neural networks [8], adaptive learning [9, 10], and the genetic algorithm (GA) [11, 12] are being used in designing such controllers. A lot of studies proposed that merged techniques provide a more accurate and robust solution than that derived from any single technique. In any practical problem, it is worth considering which should be optimized and how merging these technologies can provide an alternative to a strictly knowledge-driven reasoning system. In [13], a novel approach was proposed to represent continuous-valued input parameters using linguistic terms and then extracted fuzzy rules from trained binary single-layer neural networks. A knowledge base was learned from interval and fuzzy data for regression problems by applying the GA [14]. In addition, GA could be used to determine the membership functions in fuzzy systems [15, 16].

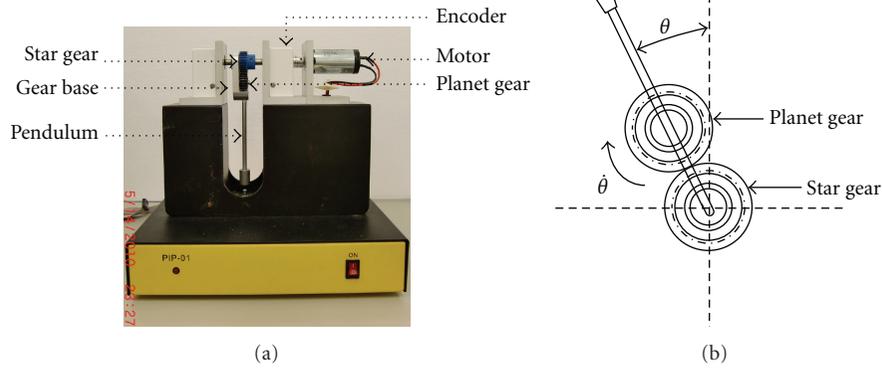


FIGURE 1: (a) Structure of the PIP system, (b) profile of the PIP system.

Scaling parameters, which describe input normalizations and output denormalization, play a role similar to that of gain coefficients in conventional PID controllers [17]. For a successful design of FLC, proper selection of input and output scaling factors is critical tasks, which in many cases, is performed through trial and error or based on some training data [18]. An interesting fuzzy controller which include seven rules was proposed to for self-tuning its scaling factors [19]. To tune the optimal parameters of the fuzzy controller at some operating points, Serra and Bottura used GA off-line to get the optimal scaling parameters [20]. Hameed et al. first tuned all these scaling parameters by the GA, and then fixed the input scaling factors and tuned the output scaling factor by the fuzzy logic [21].

This paper proposes a simple model-free control strategy for the planetary-gear-type inverted pendulum (PIP). The control strategy includes a swing-up controller and a stabilization controller, and neither of them requires a mathematical model of the PIP. For the PIP, the scaling parameters are very important and tuned by the developed GA. Rest of this paper is organized as follows. Section 2 provides a description of the PIP. Section 3 presents the smart and simple structure of the fuzzy controller. Section 4 presents the simulation and experimental results. Finally, Section 5 presents discussions and concludes the paper.

## 2. Planetary-Gear-Type Inverted Pendulum

A PIP consists of a star gear, a planet gear, an encoder, a gear base, a pendulum, and motors, as shown in Figure 1. The angular acceleration of the star gear causes the pendulum to swing, which hooks with the planetary gear. Unlike prevailing inverted pendulum systems, the PIP does not possess the winding wire problem and the length limitation of the platform [22]. The mechanical parameters are described and their values are listed in Table 1.

## 3. Control Strategy

This section presents an alternative strategy to process control using a simple model-free control in which input/output

TABLE 1: Parameters of the PIP system.

Parameter	Definition	Value
$m_{pp}$	Mass of pendulum	0.027 kg
$m_{pg}$	Mass of planetary gear	0.035 kg
$m_{sp}$	Mass of star gear	0.038 kg
$m_{pl}$	Mass of counterweight	0.02 kg
$m_{gs}$	Mass of gear base	0.95 kg
$l_{pp}$	Length of pendulum	0.11 m
$l_{pl}$	Length of payload	0.02 m
$l_{gs}$	Length of gear base	0.05 m
$J_{pp}$	Rotational inertia of pendulum	$1.54 \times 10^{-3} \text{ kg m}^2$
$J_{pg}$	Rotational inertia of planetary gear	$0.958 \times 10^{-6} \text{ kg m}^2$
$J_{sg}$	Rotational inertia of star gear	$3.211 \times 10^{-6} \text{ kg m}^2$
$r_{pg}$	Radius of planetary gear	0.013 m
$r_{sg}$	Radius of star gear	0.013 m
$G$	Acceleration of gravity	$9.8 \text{ m/s}^2$
$k_t$	Torque constant	0.242 Nm/A
$k_b$	Back-EMF constant	0.242 Nm/A
$R_a$	Armature resistance	2.21 $\Omega$

scaling parameters are optimized basing on the GA. The proposed control structure is shown in Figure 2. The strategy consists of the swing-up strategy and the fuzzy controller, neither of which has any information other than the pendulum angle and pendulum velocity.

*Remark 1.* This basic concept of the strategy with inverted pendulum involves two parts: the swing-up strategy and stabilization in the upright position method [23, 24]. Actually, along with the idea, several studies concentrated efforts on making the planetary-gear-type inverted model upright [22, 25, 26].

*Remark 2.* The adjustment of the controller gains is off-line precomputed. It may not provide any feedback to compensate for incorrect schedules [20]. Nevertheless, the

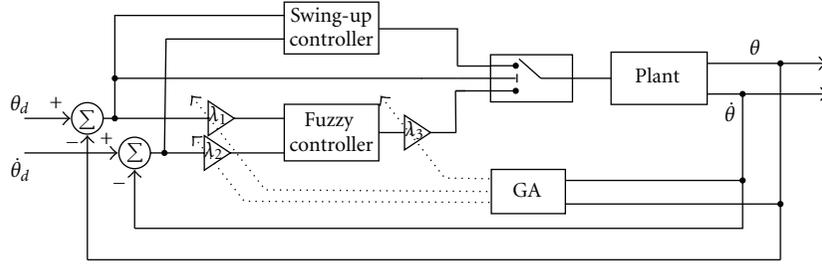


FIGURE 2: Structure of fuzzy swing-up and model-free controllers.

unchanged gain schedule may prevent instability due to frequent and rapid changes in the controller gains. In fact, the inverted pendulum is a classic example of an inherently unstable system, and it requires real-time control responses. To the best of our knowledge, the key parameters involved in genetic optimization in most studies about the inverted pendulum were computed off-line and were kept invariant during the control process.

**3.1. Swing-Up Controller.** In this study, a simple swing-up strategy is proposed. Comparing with energy-based fuzzy swing-up controller [27], the proposed method is comparatively simple and very competitive. The rules are

if  $\text{sign}(\theta) \neq \text{sign}(\dot{\theta})$ , and  $\frac{1}{2}\dot{\theta}^2 + 9.8 \times 9.81 \times (\cos(\theta) - 1) \leq 0$

then  $u = -6 \text{sign}(\dot{\theta})$ ,

if  $\text{sign}(\theta) \neq \text{sign}(\dot{\theta})$ , and  $\frac{1}{2}\dot{\theta}^2 + 9.8 \times 9.81 \times (\cos(\theta) - 1) > 0$

then  $u = 6 \text{sign}(\dot{\theta})$ , else  $u = 0$ .

(1)

### 3.2. Fuzzy Controller

**3.2.1. Fuzzy Dynamic Mode.** Let the error vector be  $E = [e_1 \ e_2]$ ,  $e_1 = \theta_d - \theta$ ,  $e_2 = \dot{\theta}_d - \dot{\theta}$ , where  $\theta_d$  is the desired angle. Using the error vector as the input, the output of the fuzzy controller is represented as a function,  $y = f(e_1, e_2)$ .

**3.2.2. Fuzzy Rule Base.** The function  $f$  is in general a complex nonlinear relationship between the inputs and the output, and it is expressed as a fuzzy rule base consisting of the following rules:

If  $e_1$  is  $A_i$  and  $e_2$  is  $B_j$ , then  $y$  is  $G_{ij}$ , where

$i = 1, 2, \dots, 7$  and  $j = 1, 2, \dots, 7$ .

**3.2.3. Membership Function and Defuzzifier.** The membership function is used to describe the uncertainty and

imprecise information. The membership value of input  $e_1$  is evaluated by  $u_{1i}$  as follows:

$$u_{1i}(x) = \begin{cases} \frac{x - A_{i-1}}{A_i - A_{i-1}}, & A_{i-1} \leq x \leq A_i, \\ \frac{x - A_{i+1}}{A_i - A_{i+1}}, & A_i \leq x \leq A_{i+1} \quad i = 2, 3, \dots, 6, \\ 0, & \text{else,} \end{cases}$$

$$u_{11}(x) = \begin{cases} 1, & x \leq A_1, \\ \frac{x - A_2}{A_1 - A_2}, & A_1 \leq x \leq A_2, \\ 0 & \text{else,} \end{cases} \quad (2)$$

$$u_{17}(x) = \begin{cases} 1, & x \geq A_7, \\ \frac{x - A_6}{A_7 - A_6}, & A_6 \leq x \leq A_7, \\ 0, & \text{else.} \end{cases}$$

The triangular input membership function is depicted in Figure 3.

Another input membership function of  $e_2$  is defined similar to that of  $e_1$  with parameters  $B_j$ ,  $i = 1, 2, \dots, 7$ .

For simplicity, a singleton fuzzification is used, as depicted in Figure 4. The center-of-gravity defuzzification method is adopted [28]. Then, the fuzzy controller function can be represented as follows:

$$f(e | \theta) = \lambda_3 \frac{\sum_{i=1}^7 \xi_i \sum_{j=1}^7 u_{1i}(\lambda_1 e_1) u_{2j}(\lambda_2 e_2)}{\sum_{j=1}^7 \sum_{i=1}^7 u_{1i}(\lambda_1 e_1) u_{2j}(\lambda_2 e_2)}. \quad (3)$$

In (3),  $\lambda_1$ ,  $\lambda_2$ , and  $\lambda_3$  are the parameters that will be optimized by GA in the subsequent step.

**3.3. Design of Input/Output Scaling Factors.** As mentioned above, fuzzy systems design is composed of three important components, the membership functions, scaling factors, and rule base. There is no standard and systematic method to adjust the shape, the parameters of the membership function, and the rule base to achieve some desired performance. However, modification of rule base may cause considerable step changes in the shape of control surface, and modification of membership functions shape can cause only local changes in the shape of control surface [28]. Additionally, modification of many parameters at a time easily results in big computation cost. So, the definition of membership functions and the establishment of control rules

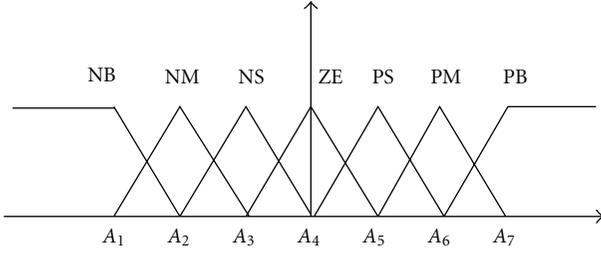


FIGURE 3: Input membership function.

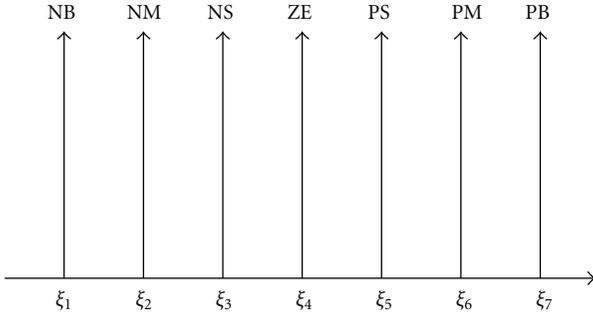


FIGURE 4: Output membership function.

base are usually designed subjective [29]. Different experts maybe obtain different experiences. It is difficult to acquire good control performance for the system whose scaling factors are just totally obtained from experts experience. In many cases, the adjusting (heuristic tuning) of scaling factors is done through trial and error or based on some training data. As global heuristic search method, genetic algorithms were used for scaling factors, which significantly simplifies the choice of the controller scaling factors for the defined control index [28].

Along with the idea, the input/output scaling gains are optimized by the GA, which was first proposed by Holland and was inspired by natural population genetics to evolve solutions to problems [30]. It consists of a number of biologically inspired steps, as shown in Figure 5. A number of approaches are available for implementing each of these steps.

**3.3.1. Coding Strategy.** The most widely used encoding method for classifiers is standard binary mapping. For the problem under consideration, each chromosome is represented by an  $n_1 + n_2 + n_3$ -bit-long chromosome, which comprises three decision variables that include the input/output gains. Each design variable is designed as

$$\text{var}_{\text{dec}}(i) = \frac{\sum_{j=1}^{n_i} 2^{\text{var}_{\text{bin}}(j)}}{2^n - 1} (\text{var}_{\text{dec}}^{L_i} - \text{var}_{\text{dec}}^{S_i}), \quad i = 1, 2, 3 \quad (4)$$

in which  $\text{var}_{\text{dec}}^{L_i}$  and  $\text{var}_{\text{dec}}^{S_i}$  are the upper and lower bound decimal values of the design variables, respectively,  $\text{var}_{\text{bin}}(j)$  is the  $j$ th element of the parameter binary vector, and  $\text{var}_{\text{dec}}(i)$  is the corresponding decimal value of the design variable. Table 2 lists the parameters of the coding strategy, selected by experience (other selections might be possible).

TABLE 2: Parameters of the coding.

Variable	Length chromosome	Lower bound	Upper bound
$\lambda_1$	$n_1 = 15$	$S_1 = -50$	$L_1 = 50$
$\lambda_2$	$n_2 = 20$	$S_2 = -500$	$L_2 = 500$
$\lambda_3$	$n_3 = 40$	$S_3 = -1000$	$L_3 = 1000$

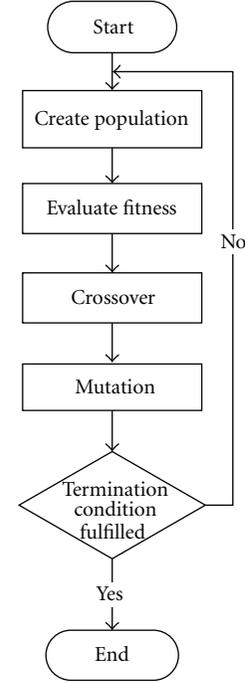


FIGURE 5: Flowchart of genetic algorithm method.

**3.3.2. Fitness Evaluation.** The fitness of a control design problem is a scalar measure of the overall performance of the controller, which indicates the quality of the solution that the chromosome values lead to. Based on these fitness values, the chromosomes that will be used to form the new generation are selected. The proper design strategy of the fitness function can pull the current state toward the desired state quickly and does not require too much computation time. The objective is to force the tracking errors to zero. Therefore, the fitness function is chosen as

$$f = -\frac{1}{P} \sum_{i=N-P}^N \sqrt{e_1^2(i) + \alpha e_2^2(i)}, \quad (5)$$

where  $e_1(i)$  and  $e_2(i)$  denote the errors of the pendulum angle and angular velocity of the pendulum for the  $i$ th training sample, respectively;  $N$  is the total number of training samples;  $P$  is the number of calculated training samples, which satisfies  $P < N$ ;  $\alpha$  is a weighting factor.

**Remark 3.** Most studies refer to the design of fitness functions based on the sum of errors between the current state and the desired state. However, for PIP control, the final steps of the states are more important. To achieve a balance between the computation time and accuracy, we adopted the above-mentioned method. In addition, Huang et al. believed

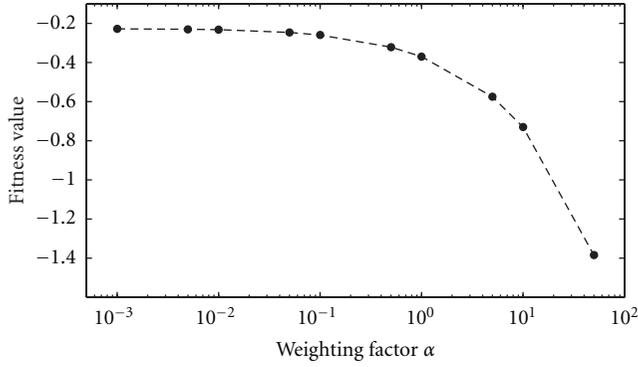


FIGURE 6: The relationship between the weighting factor and fitness value.

TABLE 3: Parameters of the GA.

Population size	100
Number of generations	400
Total number of training samples, $N$	100
Calculated number of training samples, $P$	10
Selected numbers with highest fitness values, $S_1$	2
Selected numbers of chromosomes chosen randomly, $S_2$	2
Crossover probability $p_c$	0.8
Mutation probability $p_m$	0.05
Weighting factor $\alpha$	0.01

that at different stages of the GA, the individual fitness function needs to be expanded or reduced, incorporating the nonlinear transformation for the fitness function [31].

**3.3.3. Genetic Operators.** To select individuals with high fitness to produce new individuals for the next population, the selection strategy using the following steps.

*Step 1.* Select  $s_1$  numbers of chromosomes with the highest fitness values.

*Step 2.* Select  $s_2$  numbers of chromosomes randomly based on a constructed random table and choose the one with the highest fitness value among  $s_1$  chromosomes.

**3.3.4. Crossover and Mutation.** Crossover refers to information exchange between individuals in a population in order to produce new individuals. We adopted the standard single crossover method, which takes two input individuals, selects a random point with a probability  $p_c$ , and exchanges the subindividuals behind the selected point. Mutation is traditionally performed in order to increase the diversity of the genetic information. The local maxima can be avoided. The bitwise mutation method for changing a single element is implemented with a probability  $p_m$ . Table 3 lists the parameters of the GA over the problem configuration described above.

*Remark 4.* Many experts devoted efforts to determining the relationship between the fitness and the mentioned

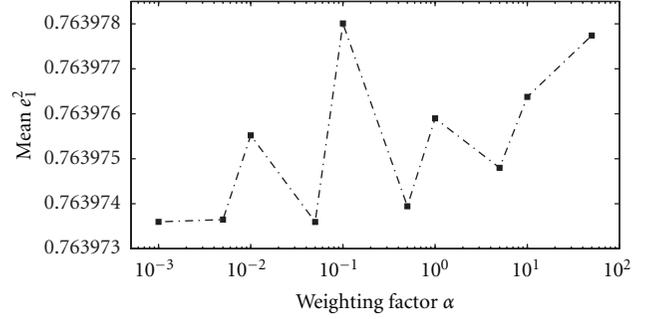


FIGURE 7: The relationship between the weighting factor and the mean-squared errors.

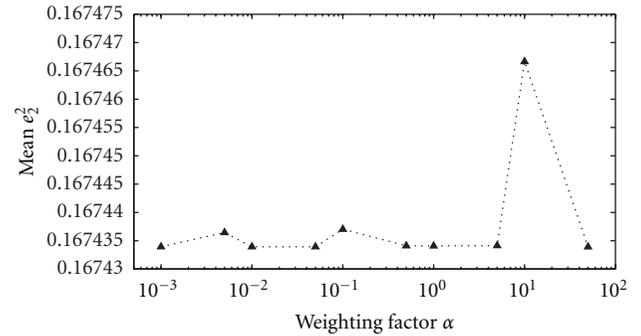


FIGURE 8: The relationship between the weighting factor and the mean-squared derivative errors.

parameters [32–34]. It is found that the larger the population size is, the higher the fitness value will be. The higher the value of  $p_c$  is, the quicker the new solutions will introduce into the population. As  $p_c$  increases, a solution can be disputed faster than selection can exploit them. Typical values of  $p_c$  are in the range of 0.5–1.0. For the parameter  $p_m$ , large values will transform the GA into a purely random search algorithm. Small values may cause the premature convergence of the GA to suboptimal solutions. Typically, the value of  $p_m$  is chosen to be in the range of 0.005–0.1.

*Remark 5.* According to (5), we can see the fitness function based on the combination of the mean-squared error and the mean-squared derivative error with a weighting factor. The choice of the weighting factor during the experimental design dose affects the final decisions. As shown in Figure 6, with the increasing weighting factor, the fitness value is increasing. However, the relationship between the mean-squared errors and the mean-squared derivative errors seems not to be very regularly in Figures 7 and 8. It looks like that when the weighting factor  $\alpha \leq 0.05$ , it offers the smaller compromise of the mean-squared errors and the mean-squared derivative errors, we choose the  $\alpha = 0.01$  as the weighting factor value.

## 4. Experimental Test

The behavior of the proposed model-free controller was investigated on a physical device. The hardware-in-the-loop controller consists of an RT-DAC/PCI motion card (which is

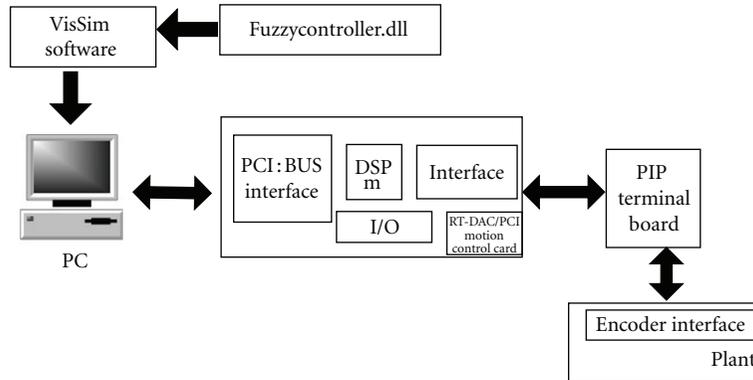


FIGURE 9: Experimental setup for PIP control system model.

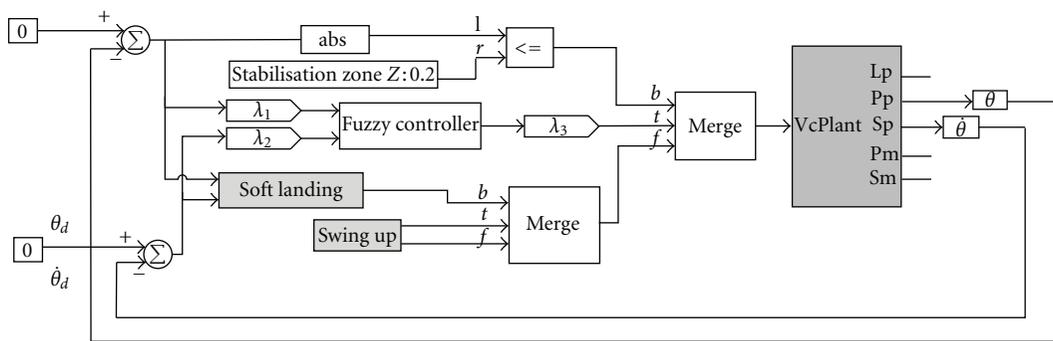


FIGURE 10: Control system interface of the VisSim interface.

inserted in the PC directly), PIP terminal card, and VisSim software. Combined with the C function library and the Windows dynamic link library (DLL), the motion controller uses the PC as its host and communicates information by PCI104 versions of BUS. Our experimental steps are as follows.

*Step 1.* Optimize the input/output gains of the fuzzy controller using Matlab.

*Step 2.* Use the C programming language and DLL in Windows environment to generate the corresponding fuzzy-controller.dll file.

*Step 3.* Input the fuzzycontroller.dll file into the VisSim software to control the PIP plant, as shown in Figure 9. The corresponding block (termed the fuzzycontroller block) is generated, as shown in Figure 10.

In this experimental application, the pendulum angle  $\theta$  can be measured by an encoder in the plant, and the pendulum angle velocity  $\dot{\theta}$  can be measured by the VisSim software. Based on the two states, the PIP terminal board finally outputs the command voltage to the dc motor of the plant. The control system interface of the VisSim interface is shown in Figure 10.

The integration computation is implemented by the Runge-Kutta method. The scaling factors are  $\lambda_1 = 44.9939$ ,  $\lambda_2 = 5.7387$ , and  $\lambda_3 = -70.5739$  with the highest

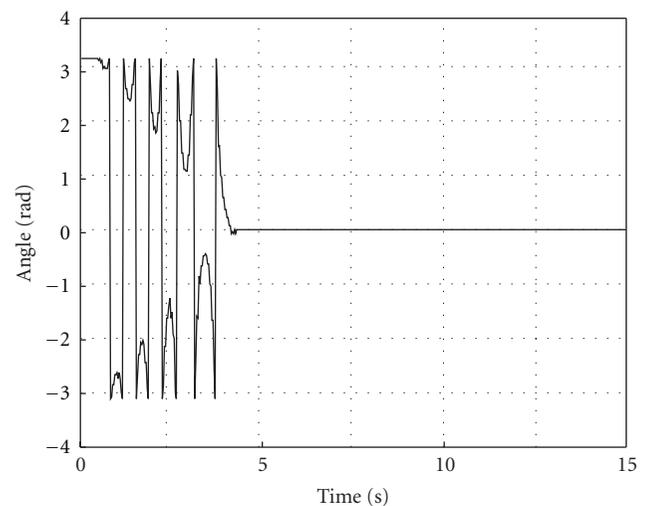


FIGURE 11: Angle of the pendulum.

fitness value. Finally, the state trajectory about the angle of the pendulum and the angular velocity of the pendulum are shown in Figures 11 and 12. The corresponding control signal of the simple model-free controller system is shown in Figure 13. The pendulum swung up in 5 seconds remains stationary. The control is indeed effective. In addition, our experiments performed in this study have been saved as videos and uploaded on YouTube. The first video demonstrates that unless the power is switched off,

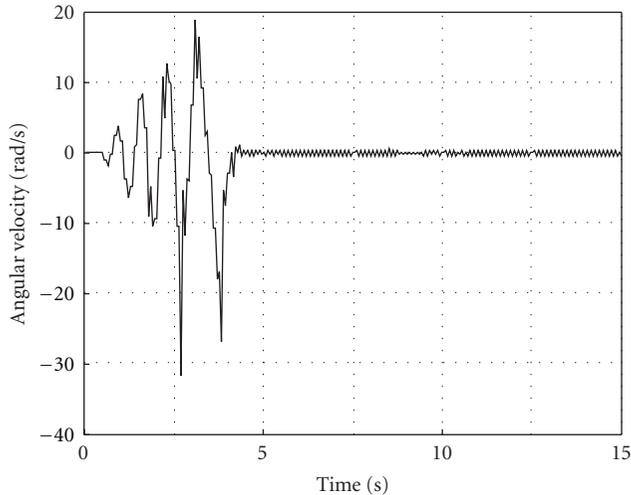


FIGURE 12: Angular velocity of the pendulum.

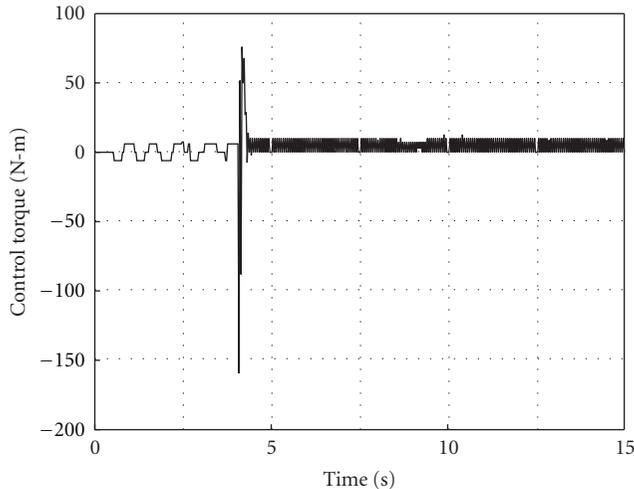


FIGURE 13: Control torque.

the pendulum can remain stationary for as long as possible ([http://www.youtube.com/watch?v=v\\_k.43Q9QVY](http://www.youtube.com/watch?v=v_k.43Q9QVY)). The second video demonstrates that the pendulum can remain upright after receiving stick punches ([http://www.youtube.com/watch?v=HCD\\_pnwR7g4](http://www.youtube.com/watch?v=HCD_pnwR7g4)). The third video demonstrates that the inverted pendulum will return to the upright position from arbitrary positions ([http://www.youtube.com/watch?v=5roOm\\_DXBS8](http://www.youtube.com/watch?v=5roOm_DXBS8)). All of the videos illustrate the effectiveness of the simple model-free controller; the second and third videos in particular also validate the robustness of the controller.

*Remark 6.* As mentioned in Remark 1, there are several efforts devoted to the PIP [22, 26]. The sliding mode control technology is applied [22, 25]. The control scheme in reference [26] consisted in fuzzy swing-up controller, fuzzy sliding balance controller, and fuzzy energy compensation mechanism [26]. However, all of these controllers were devised by preknown model knowledge with Lyapunov stability theory. The model-free controller in this paper did

not involve any preknown model knowledge. It regarded the nonlinear rotation dynamic behavior with uncertain disturbance as a whole process. Without preknown model knowledge and complicated designing progress under mathematic theory, the simple model-free controller maybe gain more extensive use. Unfortunately, as other controllers with GA optimization [11–16], it lacks instrict mathematic theory support especially convergence stability proof. It is an open range for more exploration to mathematic theory about the controller based on GA optimization algorithm in the future.

## 5. Conclusion

In this paper, a simple model-free controller for a PIP is designed. It consists of a swing-up controller and a fuzzy controller, neither containing any information about the plant. This is why we termed it a model-free strategy. The input/output scaling parameters of the controller are optimized by using the GA, which significantly simplifies the choice of these parameters for the defined controller. The experimental results, shown in the figures and videos, demonstrate the robustness and effectiveness of the strategy.

## References

- [1] R. Aguilar-López and R. Martínez-Guerra, “Control of chaotic oscillators via a class of model free active controller: suppression and synchronization,” *Chaos, Solitons and Fractals*, vol. 38, no. 2, pp. 531–540, 2008.
- [2] L. D. S. Coelho, M. W. Pessôa, R. Rodrigues Sumar, and A. Augusto Rodrigues Coelho, “Model-free adaptive control design using evolutionary-neural compensator,” *Expert Systems with Applications*, vol. 37, no. 1, pp. 499–508, 2010.
- [3] G. Qi, Z. Chen, and Z. Yuan, “Model-free control of affine chaotic systems,” *Physics Letters, Section A*, vol. 344, no. 2–4, pp. 189–202, 2005.
- [4] M. Marseguerra, E. Zio, and F. Cadini, “Genetic algorithm optimization of a model-free fuzzy control system,” *Annals of Nuclear Energy*, vol. 32, no. 7, pp. 712–728, 2005.
- [5] H. Hagrass, V. Callaghan, M. Colley, and G. Clarke, “A hierarchical fuzzy-genetic multi-agent architecture for intelligent buildings online learning, adaptation and control,” *Information Sciences*, vol. 150, no. 1–2, pp. 33–57, 2003.
- [6] C. C. Chung, H. H. Chen, and C. H. Ting, “Grey prediction fuzzy control for pH processes in the food industry,” *Journal of Food Engineering*, vol. 96, no. 4, pp. 575–582, 2010.
- [7] R. Coban and B. Can, “A trajectory tracking genetic fuzzy logic controller for nuclear research reactors,” *Energy Conversion and Management*, vol. 51, no. 3, pp. 587–593, 2010.
- [8] D. Ligas and A. Ali, “Neural net—fuzzy logic rules mapping for dynamic of fuzzy sets boundaries,” *Computers and Industrial Engineering*, vol. 31, no. 1–2, pp. 429–433, 1996.
- [9] T. C. Lin, “Observer-based robust adaptive interval type-2 fuzzy tracking control of multivariable nonlinear systems,” *Engineering Applications of Artificial Intelligence*, vol. 23, no. 3, pp. 386–399, 2010.
- [10] S. Labiod and T. M. Guerra, “Adaptive fuzzy control of a class of SISO nonaffine nonlinear systems,” *Fuzzy Sets and Systems*, vol. 158, no. 10, pp. 1126–1137, 2007.
- [11] P. M. Pawar and R. Ganguli, “Genetic fuzzy system for damage detection in beams and helicopter rotor blades,” *Computer*

- Methods in Applied Mechanics and Engineering*, vol. 192, no. 16–18, pp. 2031–2057, 2003.
- [12] H. Ishibuchi and T. Yamamoto, “Fuzzy rule selection by multi-objective genetic local search algorithms and rule evaluation measures in data mining,” *Fuzzy Sets and Systems*, vol. 141, no. 1, pp. 59–88, 2004.
- [13] S. H. Huang and H. Xing, “Extract intelligible and concise fuzzy rules from neural networks,” *Fuzzy Sets and Systems*, vol. 132, no. 2, pp. 233–243, 2002.
- [14] L. Sánchez, I. Couso, and J. Casillas, “Genetic learning of fuzzy rules based on low quality data,” *Fuzzy Sets and Systems*, vol. 160, no. 17, pp. 2524–2552, 2009.
- [15] A. Arslan and M. Kaya, “Determination of fuzzy logic membership functions using genetic algorithms,” *Fuzzy Sets and Systems*, vol. 118, no. 2, pp. 297–306, 2001.
- [16] N. R. Cazarez-Castro, L. T. Aguilar, and O. Castillo, “Fuzzy logic control with genetic membership function parameters optimization for the output regulation of a servomechanism with nonlinear backlash,” *Expert Systems with Applications*, vol. 37, no. 6, pp. 4368–4378, 2010.
- [17] A. V. Patel and B. M. Mohan, “Analytical structures and analysis of the simplest fuzzy PI controllers,” *Automatica*, vol. 38, no. 6, pp. 981–993, 2002.
- [18] R. K. Mudi and N. R. Pal, “A note on fuzzy PI-type controllers with resetting action,” *Fuzzy Sets and Systems*, vol. 121, no. 1, pp. 149–159, 2001.
- [19] H. Y. Chung, B. C. Chen, and J. J. Lin, “A PI-type fuzzy controller with self-tuning scaling factors,” *Fuzzy Sets and Systems*, vol. 93, no. 1, pp. 23–28, 1998.
- [20] G. L. O. Serra and C. P. Bottura, “Multiobjective evolution based fuzzy PI controller design for nonlinear systems,” *Engineering Applications of Artificial Intelligence*, vol. 19, no. 2, pp. 157–167, 2006.
- [21] S. Hameed, B. Das, and V. Pant, “A self-tuning fuzzy PI controller for TCSC to improve power system stability,” *Electric Power Systems Research*, vol. 78, no. 10, pp. 1726–1735, 2008.
- [22] T. C. Kuo, Y. J. Huang, and S. H. Chang, “Sliding mode control with self-tuning law for uncertain nonlinear systems,” *ISA Transactions*, vol. 47, no. 2, pp. 171–178, 2008.
- [23] M. Alamir and A. Murilo, “Swing-up and stabilization of a Twin-Pendulum under state and control constraints by a fast NMPC scheme,” *Automatica*, vol. 44, no. 5, pp. 1319–1324, 2008.
- [24] D. Chatterjee, A. Patra, and H. K. Joglekar, “Swing-up and stabilization of a cart-pendulum system under restricted cart track length,” *Systems and Control Letters*, vol. 47, no. 4, pp. 355–364, 2002.
- [25] Y. H. Chang, C. W. Chang, J. S. Taur, and C. W. Tao, “Fuzzy swing-up and fuzzy sliding-mode balance control for a planetary-gear-type inverted pendulum,” *IEEE Transactions on Industrial Electronics*, vol. 56, no. 9, pp. 3751–3761, 2009.
- [26] Y. J. Huang, T. C. Kuo, and S. H. Chang, “Adaptive sliding-mode control for nonlinear systems with uncertain parameters,” *IEEE Transactions on Systems, Man, and Cybernetics, Part B*, vol. 38, no. 2, pp. 534–539, 2008.
- [27] K. J. Åström and K. Furuta, “Swinging up a pendulum by energy control,” *Automatica*, vol. 36, no. 2, pp. 287–295, 2000.
- [28] T. Orłowska-Kowalska, K. Szabat, and K. Jaszczak, “The influence of parameters and structure of PI-type fuzzy-logic controller on DC drive system dynamics,” *Fuzzy Sets and Systems*, vol. 131, no. 2, pp. 251–264, 2002.
- [29] H. Y. Chung, B. C. Chen, and J. J. Lin, “A PI-type fuzzy controller with self-tuning scaling factors,” *Fuzzy Sets and Systems*, vol. 93, no. 1, pp. 23–28, 1998.
- [30] J. H. Holland, *Adaptation in Natural and Artificial Systems*, University of Michigan Press, 1975.
- [31] H. Wei, Q. Qian, H. Qiang, H. Qiaoli, Z. Yixin, and X. Lin, “Optimization of sliding mode controller for double inverted pendulum based on genetic algorithm,” in *Proceedings of the 2nd International Symposium on Systems and Control in Aerospace and Astronautics (ISSCAA '08)*, pp. 1–5, 2008.
- [32] J. O. Schaffer, R. A. Caruana, L. J. Eshelman, and R. Das, “A study of control parameters affecting online performance of genetic algorithms for function optimization,” in *Proceedings of the 3rd International Conference on Genetic Algorithms*, pp. 51–60, 1989.
- [33] N. Metni, “Neuro-control of an inverted pendulum using genetic algorithm,” in *Proceedings of the International Conference on Advances in Computational Tools for Engineering Applications (ACTEA '09)*, pp. 27–33, July 2009.
- [34] A. Poursamad and M. Montazeri, “Design of genetic-fuzzy control strategy for parallel hybrid electric vehicles,” *Control Engineering Practice*, vol. 16, no. 7, pp. 861–873, 2008.

