The Thirteenth International Symposium on Artificial Life and Robotics 2008(AROB 13th '08),
B-Con Plaza, Beppu, Oita, Japan, January 31-February 2, 2008

Integration of PSO and GA for Optimum Design of Fuzzy PID Controllers in a Pendubot System

Yu-Yi Fu*, Chia-Ju Wu**, Ting-Li Chien***, and Chia-Nan Ko****

*Graduate School of Engineering Science and Technology, National Yunlin University of Science and Technology
Douliou, Yunlin 640, Taiwan

**Department of Electrical Engineering, National Yunlin University of Science and Technology
Douliou, Yunlin 640, Taiwan

***Department of Electronic Engineering, Wu-Feng Institute of Technology
Minghsiung,Chiayi 621, Taiwan

****Department of Automation Engineering, Nan-Kai Institute of Technology
Tasotun, Nantou 542, Taiwan

(email: * t098@nkc.edu.tw ; ** wucj@yuntech.edu.tw ; *** cdl@mail.wfc.edu.tw ; **** t105@nkc.edu.tw)

Abstract: In this paper, a novel auto-tuning method is proposed to design fuzzy PID controllers for asymptotical stabilization of a pendubot system. In the proposed method, a fuzzy PID controller is expressed in terms of fuzzy rules, in which the input variables are the error signals and their derivatives, while the output variables are the PID gains. In this manner, the PID gains are adaptive and the fuzzy PID controller has more flexibility and capability than the conventional ones with fixed gains. To tune the fuzzy PID controller simultaneously, an evolutionary learning algorithm integrating particle swarm optimization (PSO) and genetic algorithm (GA) methods is proposed. The simulation results illustrate that the proposed method is indeed more efficient in improving the asymptotical stability of the pendubot system.

Keywords: Particle swarm optimization, genetic algorithm, fuzzy PID controllers, pendubot system

I. INTRODUCTION

Recently, many methods have been proposed for the design of controllers. Most industrial processes nowadays are still controlled by PID controllers. However, a conventional PID controller may have poor control performance for nonlinear and/or complex systems that have no precise mathematical models. In Keel et al [1] and Cervantes et al [2], how to tune the PID controller based on mathematical models is proposed, but complex mathematical computation is generally required in tuning procedures. For the methods in Whidborne and Istepanian [3] and Lin et al [4], since the PID gains are fixed, the main disadvantage is that they usually lack in flexibility and capability.

Fuzzy controllers provide reasonable and effective alternatives for conventional controllers. Many researchers attempted to combine conventional PID controllers with fuzzy logic (Tao and Taur [5] and Wu et al [6]). Despite the significant improvement of these fuzzy PID controllers over their classical counterparts, it should be noted that they still have disadvantages. Furthermore, for nonlinear multivariable systems, how to reduce the number of fuzzy rules is unsolved. In Wu et al [6], a multivariable system is decomposed into several SISO systems in order to reduce the number of fuzzy rules. However, the drawback is that the interdependency between the variables is neglected such that the designed controller cannot work well for some cases.

Several evolutionary algorithms have been proposed recently to search for optimal PID controllers. Among them, genetic algorithm (GA) has received great attention and particle swarm optimization (PSO) method has been successfully applied to various fields (Gaing [7] and Habib and Al-kazemi [8]). In this paper, an evolutionary learning algorithm integrating PSO and GA methods will be adopted to perform the optimum design of the proposed fuzzy PID controller. In this manner, the proposed method is fully capable of creating a fuzzy PID controller and eliminates the need for human expertise information in the design process. To show the flexibility and capability of the proposed method, an underactuated pendubot system is adopted as an illustrative example. From the simulation results, one can find that the designed fuzzy PID controller is more versatile than a conventional one.

II. FUZZY PID CONTROLLERS

In a classical PID control system, the time-domain form of a PID controller is usually expressed as

$$u(t) = K_P e(t) + K_I \int e(t)dt + K_D \dot{e}(t) \qquad (1)$$

where $u(t)$ is the control signal, $e(t)$ is the error signal, and K_P, K_I, and K_D denote the proportional gain, the integral gain, and the derivative gain, respectively.

In the proposed fuzzy PID controller, the input variables of the fuzzy rules are the error signals and their derivatives, while the output variables are the PID gains. The fuzzy PID control rules are expressed as

If e_1 is X_1^i and $\dot{e}_1$ is X_2^j and e_2 is X_3^k and $\dot{e}_2$ is X_4^l,
then $K_{P1} = Y_{P1}^{ijkl}, K_{I1} = Y_{I1}^{ijkl}, \cdots, K_{D2} = Y_{D2}^{ijkl}$
for $1 \le i \le n_1$, $1 \le j \le n_2$, $1 \le k \le n_3$, $1 \le l \le n_4$ (2)

where e_1, e_2 and $\dot{e}_1$, $\dot{e}_2$ are the error signals and their derivatives, $X_1^i, X_2^j, X_3^k, X_4^l$ are the membership functions of e_1, $\dot{e}_1$, e_2, and $\dot{e}_2$, $K_{P1}, K_{I1}, \cdots, K_{D2}$ are the PID gains, $Y_{P1}^{ijkl}, Y_{I1}^{ijkl}, \cdots, Y_{D2}^{ijkl}$ are real numbers, n_1, n_2, n_3, and n_4 denote the numbers of input membership functions, respectively.

The membership functions of an FLC are usually parametric functions such as triangular functions, trapezoidal functions, Gaussian functions, and singletons. Though the proposed method is equally applicable to all these kinds of membership functions, asymmetric Gaussian ones are used as the antecedent fuzzy sets in this paper. This means that input membership functions are represented as

$$X_k^{m_k}(x_k) = \begin{cases} \exp\left[-\left(\dfrac{x_k - \rho_k^{m_k}}{\sigma_{kl}^{m_k}}\right)^2\right] & \text{if } x_k \le \rho_k^{m_k} \\ \exp\left[-\left(\dfrac{x_k - \rho_k^{m_k}}{\sigma_{kr}^{m_k}}\right)^2\right] & \text{if } x_k > \rho_k^{m_k} \end{cases}$$

$$\text{for } k = 1, 2, \cdots, 4,$$

$$1 \le m_1 \le n_1, 1 \le m_2 \le n_2, 1 \le m_3 \le n_3, 1 \le m_4 \le n_4 \quad (3)$$

where x_k represents the input linguistic variables, $\rho_k^{m_k}$, $\sigma_{kl}^{m_k}$, and $\sigma_{kr}^{m_k}$ denote the values of the centers, the left widths, and the right widths of the input membership functions, respectively. For the output membership functions, singleton sets are adopted. In the defuzzification process, Wang [9] used the center of gravity method to determine the output crisp values. Then, if the PID control law is used, the control signal is determined as

$$\begin{aligned} u(t) = K_{P1}e_1(t) + K_{I1}\int e_1(t)dt + K_{D1}\dot{e}_1(t) \\ + K_{P2}e_2(t) + K_{I2}\int e_2(t)dt + K_{D2}\dot{e}_2(t) \end{aligned} \quad (4)$$

From the above description, one can find that the gains of the fuzzy PID controller are adaptive such that the controller should have more flexibility and capability than the conventional ones. However, it is very difficult, if not impossible, to determine the parameters directly. Therefore, a novel method integrating PSO and GA is proposed to search for the optimal values of these parameters simultaneously.

III. INTEGRATION OF PSO AND GA

PSO is a population-based stochastic searching technique developed by Kennedy and Eberhart [10]. It is similar to the GA in that it begins with a random population matrix and searches for the optima by updating generations.

1. Particle Representations

Before applying the novel auto-tuning method, how to encode the parameters must be introduced firstly. In the proposed method, a mixed coding method is used, in which n_1, n_2, n_3, and n_4 are encoded as binary numbers and $\rho_k^{m_k}$, $\sigma_{kl}^{m_k}$, $\sigma_{kr}^{m_k}$, Y_{P1}^{ijkl}, Y_{I1}^{ijkl}, Y_{D1}^{ijkl}, Y_{P2}^{ijkl}, Y_{I2}^{ijkl}, Y_{D2}^{ijkl} are encoded as real numbers. This means that the positions of particles are represented as

$$P = [p_{binary} \; p_{real}] \quad (5)$$

where

$$p_{binary} = [p_1, p_2, \cdots, p_{n_1+n_2+n_3+n_4}] \quad (6)$$

$$p_{real} = [p_{n_1+n_2+n_3+n_4+1}, \cdots, p_{4n_1+4n_2+4n_3+4n_4+6n_1n_2n_3n_4}] \quad (7)$$

The particle p_{binary} contains binary variables taking the value of one or zero. The elements of p_{binary} are used to indicate which ones of the membership functions are activated. As for the real particles p_{real}, the elements of p_{real} are used to represent the values of $\rho_k^{m_k}$, $\sigma_{kl}^{m_k}$, $\sigma_{kr}^{m_k}$, Y_{P1}^{ijkl}, Y_{I1}^{ijkl}, Y_{D1}^{ijkl}, Y_{P2}^{ijkl}, Y_{I2}^{ijkl}, and Y_{D2}^{ijkl}.

2. Evolutionary Algorithms

In evolutionary strategies, the real particles p_{real} will employ the PSO method. As for binary particles p_{binary}, it will adopt the GA because of their nature and simplicity. In PSO method, the particles update their velocities and positions based on the local best and global best solutions [11]. In the evolutionary procedure, the inertia weight, the cognitive parameter, and the social parameter are linearly adaptable over the evolutionary procedure [11]. In the proposed GA-based method for binary particles, one cut-point crossover operator and single-point mutation operator will be employed (Haupt and Haupt [12]).

IV. AN SIMULATION EXAMPLE

1. The Pendubot System

The general dynamic model of underactuated mechanisms with m actuated joints from a total of n joint can be expressed as follows (Spong and Vidyasagar[13]) :

$$M(q)\ddot{q} + C(q, \dot{q})\dot{q} + G(q) = \tau \quad (8)$$

where $q \in R^n$ is the position vector indicating link angles, $M(q)$ denotes the $n \times n$ inertia matrix, $C(q, \dot{q})\dot{q} \in R^n$ is the vector of damping, coriolis, and centrifugal torques, $G(q) \in R^n$ represents the gravitational term and $\tau \in R^n$ is the vector of control torque which has $(n-m)$ zero components.

For the pendubot system in Fig. 1, let m_1 and m_2 denote the distributed mass of the actuated link (called link 1) and the unactuated link (called link 2), respectively. Meanwhile, let q_1 and q_2 denote the angles of the two links, l_1 and l_2 denote the lengths of the two links, and l_{1c} and l_{2c} denote the distances to the center of masses, respectively.

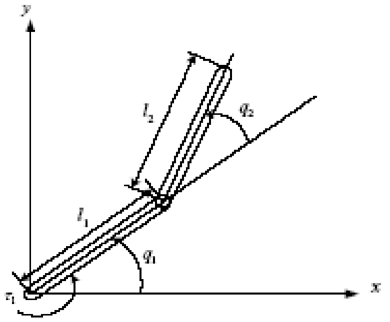


Fig. 1. Dynamics of the pendubot system

When the configuration is at equilibrium state; that is, pendubot balances at a state $\dot{q}=0$ and $\ddot{q}=0$, the following can be derived from (8).

$$(m_1 g l_{c1} + m_2 g l_1)\cos q_1 + m_2 g l_{c2}\cos(q_1+q_2) = \tau_1 \quad (9)$$

$$m_2 g l_{c2}\cos(q_1+q_2) = 0 \quad (10)$$

2. PSO-GA tuning Fuzzy PID Controller

In the pendubot system, the desired value of $q_1(t)$ and $q_2(t)$ are denoted by q_{1d} and q_{2d}. If the PID control law is employed, then the input-output relation of the pendubot system is expressed as

$$\tau(t) = k_{P1}e_1(t) + k_{I1}\int e_1(t)dt + k_{D1}\dot{e}_1(t) + k_{P2}e_2(t) + k_{I2}\int e_2(t)dt + k_{D2}\dot{e}_2(t) \quad (11)$$

where $e_1(t) = q_{1d} - q_1(t)$, $e_2(t) = q_{2d} - q_2(t)$, $\dot{e}_1(t) = \dot{q}_{1d} - \dot{q}_1(t)$, and $\dot{e}_2(t) = \dot{q}_{2d} - \dot{q}_2(t)$.

3. Fitness

In designing the fuzzy PID controller, the primary goal is to drive a pendubot system from the given initial state to the desired final state. However, if the number of fuzzy rules is large, then heavy computation burden and huge memory requirement are inevitable. Therefore, the primary goal and the way to reduce the number of fuzzy rules should be taken into account simultaneously in defining the fitness function. This means that two performance criteria can be included in the same fitness as follows:

$$f = \frac{1}{[(1+\sum_{i=1}^{n_1} p_i)\cdot(1+\sum_{j=1}^{n_2} p_j)\cdot(1+\sum_{k=1}^{n_3} p_k)\cdot(1+\sum_{l=1}^{n_4} p_l)]^2} + \frac{1}{\int t[e_1^2(t)+e_2^2(t)]dt} \quad (12)$$

where p_i, p_j, p_k, and p_l are the binary elements to indicate which ones of the membership functions are activated. From the definition (12), the fitness value can be calculated to evaluate the performance of the fuzzy PID controller and a higher fitness value denotes a better performance.

V. SIMULATION RESULTS

The parameters of the pendubot system shown in Fig. 1 are chosen as $m_1 = 2.0$ kg, $m_2 = 1.5$ kg, $l_1 = 0.3$ m, $l_2 = 0.5$ m, $l_{1c} = 0.15$ m, $l_{2c} = 0.25$ m, and $g = 9.8$ m/s^2. The initial state and the desired final state of the pendubot pendubot system are $[q_1, \dot{q}_1, q_2, \dot{q}_2] = [-\pi/2,\ 0,\ 0,\ 0]$ and $[q_1, \dot{q}_1, q_2, \dot{q}_2] = [\pi/2,\ 0,\ 0,\ 0]$. Meanwhile, the input torque $\tau(t)$ of the motor is assumed to be within the range $[-10\text{ Nm}, 10\text{ Nm}]$.

In the proposed algorithm, the population size, the maximal iteration number, the crossover rate, and mutation rate are chosen to be 20, 10000, 0.8, and 0.2, respectively. Moreover, it is assumed that the values of n_1, n_2, n_3, and n_4 are all chosen as five, and the singletons of the output linguistic variables are all chosen as real numbers in the range $[-10, 10]$. According to the procedure of the PSO-GA algorithm, the minimal fuzzy rules are determined and the optimal membership functions of the input linguistic variables are shown in Fig. 2.

To demonstrate the proposed fuzzy PID controller is superior to the conventional PID controller, one also uses the local optimal values of PID gains to search the global optimum. Corresponding to these PID gains, the plots of $q_1(t)$ and $q_2(t)$ for the pendubot system are shown in Fig. 3 and Fig. 4, respectively. By comparing the plots in Fig. 3 and Fig. 4, one can easily find that the designed fuzzy PID controller has a better performance than the fixed gains PID one.

VI. CONCLUSION

In PID tuning techniques, the PID gains are difficult to obtain the optimal values for stabilizing a pendubot system. In this paper, one presents an integrating PSO and GA approach to design a fuzzy PID controller to asymptotically stabilize the pendubot and maintain the equilibrium state over all control processes. From the simulation results, one demonstrates the proposed fuzzy PID controller has superior performance for asymptotical stabilization of the pendubot system.

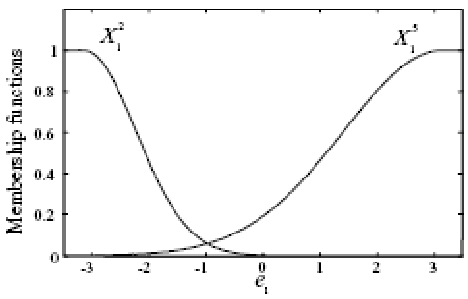
ACKNOWLEDGMENT

This work was supported in part by the National Science Council, Taiwan, R.O.C., under grant NSC96-2628-E-224-007-MY2.

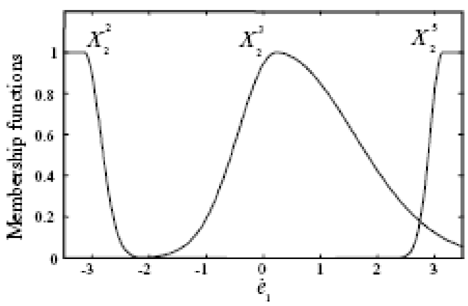
REFERENCES

[1] Keel LH, Rego JI, Bhattacharyya SP (2003), A new approach to digital PID controller design. IEEE Trans on Automatic Control 48(4):687-692
[2] Cervantes I, Garrido R, Jose AR, et al (2004), Vision-based PID control of planar robots. IEEE Trans on Mechatronics 9(1):132-136

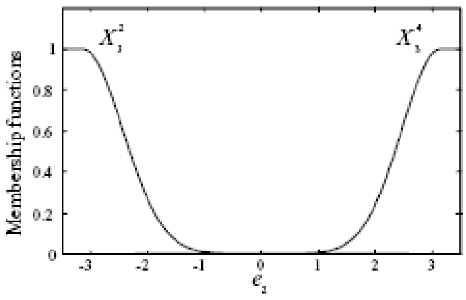
[3] Whidborne JF, Istepanian RSH (2001), Genetic algorithm approach to designing finite-precision controller structures, IEE Proc of Control Theory Applications 148(5):377-382
[4] Lin L, Jan HY, Shieh NC (2003), GA-based multiobjective PID control for a linear brushless dc motor. IEEE Trans on Mechatronics 8(1):56-65
[5] Tao CW, Taur JS (2005), Robust fuzzy control for a plant with fuzzy linear model. IEEE Trans on Fuzzy Syst 13(1):30-41
[6] Wu CJ, Liu GY, Cheng MY, et al (2002), A neural-network-based method for fuzzy parameter tuning of PID controllers. J of the Chin Inst of Eng 25(3): 265-276
[7] Gaing ZL (2004), A particle swarm optimization approach for optimum design of PID controller in AVR system. IEEE Trans on Energy Conversion 19(2): 384-391
[8] Habib SJ, Al-kazemi BS (2005), Comparative study between the internal behavior of GA and PSO through problem-specific distance functions. The 2005 IEEE Congress on Evolutionary Computation, pp. 2190- 2195
[9] Wang LX (1997), A Course in Fuzzy Systems and Control. Prentice-Hall, New Jersey
[10] Kennedy J, Eberhart R (1995), Particle swarm optimization. Proceedings of IEEE International Conference on Neural Networks, pp. 1942-1948
[11] Ratnaweera A, Halgamuge SK, Watson C (2004), Self-organizing hierarchical particle swarm optimizer with time-varying acceleration coefficients. IEEE Trans on Evolutionary Comput 8(3):240-255
[12] Haupt RL, Haupt SE (2004), Practical Genetic Algorithms, 2nd edition. John Wiley & Sons, New York
[13] Spong MW, Vidyasagar M (1989), Robot dynamics and control. Wiley, New York



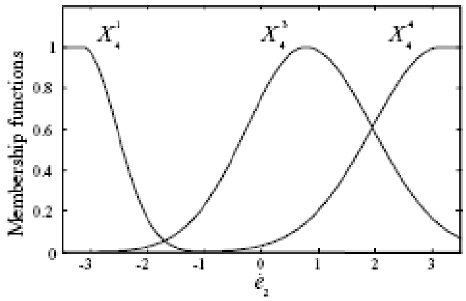
(a) Optimal membership functions of $e_1(t)$.



(b) Optimal membership functions of $\dot{e}_1(t)$.



(c) Optimal membership functions of $e_2(t)$.



(d) Optimal membership functions of $\dot{e}_2(t)$.

Fig. 2. The optimal membership functions of the input linguistic variables.

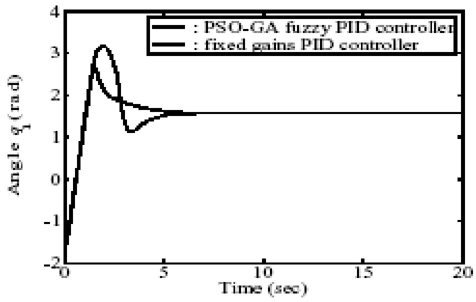


Fig. 3. Plots of angle $q_1(t)$ of the pendubot system, which are generated by the PSO-GA fuzzy PID controller and a fixed gains PID controller, respectively.

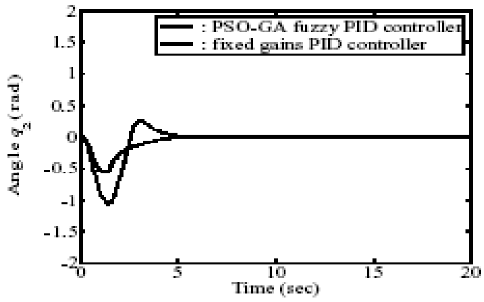


Fig. 4. Plots of angle $q_2(t)$ of the pendubot system, which are generated by the PSO-GA fuzzy PID controller and a fixed gains PID controller, respectively.