ADAPTIVE-PID EXPERIMENTAL STM32F4 CONTROLLER FOR ROTARY INVERTED PENDULUM

Le-Anh-Dung Pham[1, 2]*, Vu-Quan Tran[2], Thanh-Tay Nguyen[2], Minh-Huy Le[2], Viet-Duc Nguyen[2], Van-Hiep Le[2], Thanh-Thai Pham[2], Tan-Dai Trinh[2]

[1] Bosch Global Software Technologies
Etown 2, Cong Hoa street, 364, Ho Chi Minh city, Vietnam

[2] Ho Chi Minh city University of Technology and Education (HCMUTE)
Vo Van Ngan street, 01, Ho Chi Minh city, Vietnam

***** Corresponding author. E-mail: Dung.PhamLeAnh@vn.bosch.com

Abstract: Rotary inverted pendulum (RIP) is a classical model of control engineering. Paper deals with a PID-adaptive structure which is based on structure of neuron to train Kp, Ki, Kd through operation. In simulation, our adaptive controller is proven to work in larger range than classical PID controller. Through experimental model using STM32F4, we prove vibration of system under adaptive-PID is smaller than under classical PID structure. Then, combination of neuron network (NN) and PID control can be used as simple structure for single-input multi output (SIMO) systems which are similar to RIP.

Keywords: PID control, adaptive control, inverted pendulum, STM32F4, SIMO system.

1. Introduction

In nonlinear systems, RIP is easy to be built and has the most basic nonlinear characteristics. Therefore, above system is a common object for identification and control experiments. Different control algorithms have been applied to RIP model such as PID, pole-placement, LQR, fuzzy control, NNs and it achieved considerable success.

Authors in [1] proposed a suitable pole selection method and successfully simulated the pole-placement set controller to control RIP. Quanser [2] also built a RIP and applied LQR control for theory training. Above control techniques require designers to have good knowledge of the mathematical model of system and experiences in selecting the appropriate parameters for stable control. Authors in [3] combined two sets of PID to control one input - one output to get a compromised controller, it could be used for controlling one input - multiple output systems and successfully applied it to the RIP. The above technique does not require the designer to have good knowledge of the mathematical model of the system. Controller designers need experience and trial-and-error time to get the best controller. However, the system only operates well around the static work point. Authors in [4] mentioned the method of using neuron-PID controllers. The combination of the PID controller with the neural structure allows PID parameters to gradually change to reach the most optimal value. However, in this paper, the object used is a linear object, one input – one output, and the successful result stops at the simulation.

In this paper, we proposed using a PID-adaptive set in combination with a static two-variable PID at [3] to control RIP. PID-adaptive controllers are applied to single input - multiple output (SIMO) systems and do not need to have good knowledge of the mathematical model of system. Ability to self-tuning online of NN gradually adjusts Kp, Ki, Kd parameters in the direction of optimization.

2. Mathematical Model of RIP

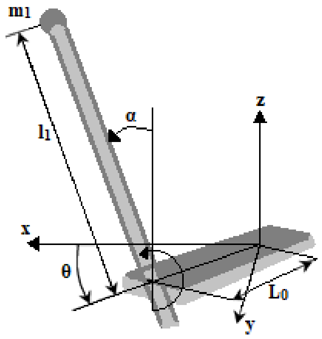

Fig. 1. Mathematical model of RIP

System of equations describing the nonlinear dynamic characterization of RIP (as shown in Fig. 1) is

$$A\begin{bmatrix}\ddot{\theta}\\ \ddot{\alpha}\end{bmatrix} + B\begin{bmatrix}\dot{\theta}\\ \dot{\alpha}\end{bmatrix} + C = D_u \quad (1)$$

where:

$$A = \begin{bmatrix} J_0 + m_1 L_0{}^2 + m_1 L_1{}^2 \sin^2\alpha & -m_1 L_0 l_1 \cos\alpha \\ m_1 L_0 l_1 \cos\theta & J_1 + m_1 l_1{}^2 \end{bmatrix};$$

$B =$
$$\begin{bmatrix} C_0 + \frac{K_t K_b}{R_a} + \frac{1}{2} m_1 {l_1}^2 \dot{\alpha} \sin 2\alpha & m_1 L_0 l_1 \dot{\alpha} \sin \alpha + \frac{1}{2} m_1 {l_1}^2 \dot{\theta} \sin 2\alpha \\ -\frac{1}{2} m_1 {l_1}^2 \dot{\theta} \sin 2\alpha & C_1 \end{bmatrix}$$
$$C = \begin{bmatrix} 0 \\ -m_1 g l_1 \sin \alpha \end{bmatrix}; D = \begin{bmatrix} \frac{K_t K_b}{R_a} \\ 0 \end{bmatrix}$$

where θ: Arm rotation angle (rad); K_t: Motor torque constant (Nm/A); K_b: Generic electrical constant. (Vs/rad); C_1: Coefficient of sliding friction of the pendulum (kgm^2/s); K_u: Motor control signal amplification coefficient (V/count); R_a: Rotor resistance of the motor (Ω); u: Motor control signal (V); g: Gravitational acceleration (m/s^2); α: Angle of movement of pendulum (rad); J_0: Moment of inertia of arm (kgm^2); J_1: Moment of inertia of pendulum (kgm^2).

According to [3], in order to control SIMO system such as RIP, the above authors proposed a static two-variable PID control with a stable pendulum control block diagram as Fig. 2. The two PID controllers control pendulum angle α and arm angle θ, respectively. The control signal is the result of the compromise of these two controllers.

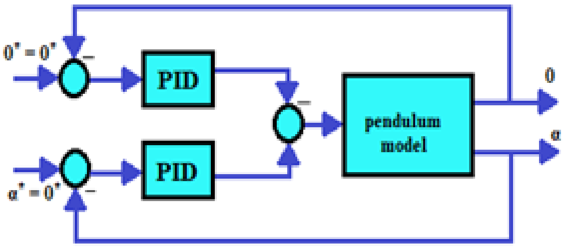

Fig. 2. Static two-variable PID control diagram controlling RIP

3. PID-Adaptive Controller for RIP

NN is a simple mathematical model of the human brain. It is composed of neurons connected by links. Each link is companied by a weight, which characterizes excitability or inhibition between neurons [5]. Fig. 3 describes a diagram of an artificial neuron, where x_1, x_2,…x_m is the signal to neurons and w_1, w_2,…w_m is the weight of neurons.

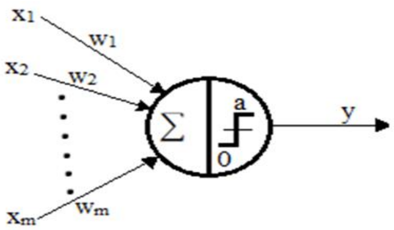

Fig. 3. Block diagram of an artificial neuron

The information processing of NN is divided into two parts: input processing (aggregate function) and output processing (impact function). When NN is trained, the weights w_1, w_2,…w_m will change in turn to achieve the most optimal value. There are three main methods of NN training: batch training, online training, stochastic training [6].

Inverted pendulum object has a fast impact time. The method of randomly selecting weights can make the pendulum system unbalanced and unstable, the method of stochastic training is inappropriate. On the other hand, the neural network weight must be constantly updated with the initial x-state input number being deterministic. Batch training is inappropriate. From there, the authors choose an online training method to train the control weights for RIP.

In this paper, we use a direct control algorithm. Parameters K_P, K_I, K_D correspond to the weight of NN consisting of a neuron. The self-tuning NN structure is shown in Fig. 4.

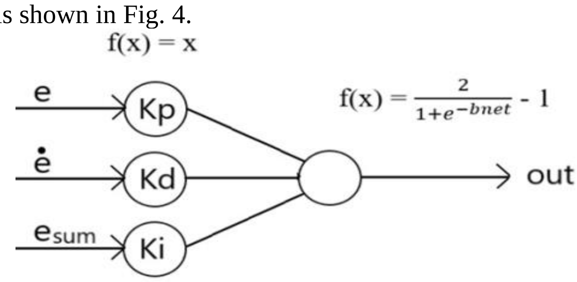

Fig. 4. PID-adaptive controller

Input aggregation function:

$$u = K_P e + K_I e_{sum} + K_p \dot{e} \qquad (2)$$

The impact function at the output selected is the dipole S-function:

$$out = h(net) = \frac{1-e^{-bnet}}{1+e^{-bnet}} \qquad (3)$$

Thus, output of neuron is a differentiable tansig function suitable for performing its own derivative calculations.

Selected target function:

$$J = \frac{1}{2}(y_d - y)^2 \qquad (4)$$

where y_d is the desired output value and y is the actual output value.

The weight is updated according to the gradient (steepest descent) method, that is:

$$\theta(k+1) = \theta(k) - \eta \nabla J \qquad (5)$$

where $\eta > 0$ is the constant, which affects the learning speed and convergence of the neural network weight.

Parameters of PID controller change according to the rule:

$$\begin{cases} K_P = K_P - \eta \frac{\partial E}{\partial K_P} \\ K_I = K_I - \eta \frac{\partial E}{\partial K_I} \\ K_D = K_D - \eta \frac{\partial E}{\partial K_D} \end{cases} \qquad (6)$$

Conducting separate derivative analysis of expressions Eq. (6):

$$\begin{cases} \frac{\partial E}{\partial K_P} = \frac{\partial E}{\partial y}\frac{\partial y}{\partial u}\frac{\partial u}{\partial K_P} = (-e)f'(u)e \\ \frac{\partial E}{\partial K_I} = \frac{\partial E}{\partial y}\frac{\partial y}{\partial u}\frac{\partial u}{\partial K_I} = (-e)f'(u)e_{sum} \\ \frac{\partial E}{\partial K_D} = \frac{\partial E}{\partial y}\frac{\partial y}{\partial u}\frac{\partial u}{\partial K_D} = (-e)f'(u)\dot{e} \end{cases} \quad (7)$$

Law of changeness of weights of PID-adaptive controller is:

$$\begin{cases} K_P = K_P - \eta e f'(u) e \\ K_I = K_P - \eta e f'(u) e_{sum} \\ K_D = K_D - \eta e f'(u) \dot{e} \end{cases} \quad (8)$$

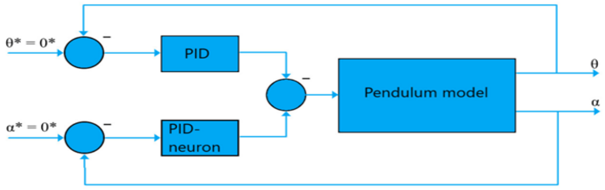

Fig. 5. PID-adaptive coupled two-variable PID control diagram that controls RIP

4. Simulation Results

Initial parameters of selected pendulum angle controller is $K_P = 10$, $K_I = 0.1$, $K_D = 1$. The selected arithmetic η is 0.01. Fig. 6 shows a comparison of the output response of RIP when a static PID is applied (K_P, K_I, K_D constant during operation) and PID sets with neuron structure combinations (K_P, K_I, K_D gradually change in the direction of optimization). The pendulum deflection angle α gradually increase in the case of applying a static PID controller, causing the system to destabilize. In the case of applying a PID-adaptive controller, after a period of time, the system remains stable with a pendulum deflection angle of 0.

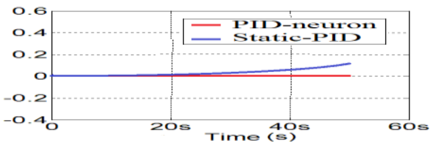

Fig. 6. Angle of pendulum (rad) under static PID controller and PID-adaptive controller

Process of changing parameters K_P, K_I, K_D of PID-adaptive regulator is shown in Fig. 7, Fig. 8 and Fig. 9, respectively. After a period of 0.2s, PID parameter value is determined with $K_P = 1122.5$, $K_I \approx 0.112$, $K_D = 39.6$ and no longer changes significantly.

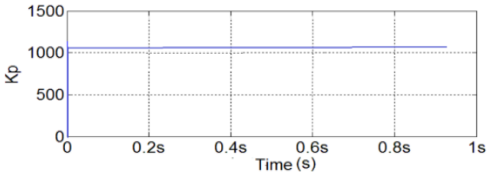

Fig. 7. Process of changing Kp parameter.

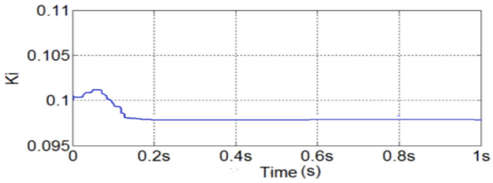

Fig. 8. Process of changing Ki parameter

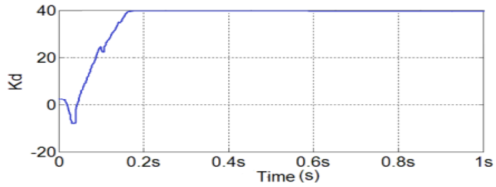

Fig. 9. The process of changing Kd parameter

5. Experimental Results

We utilize an experimental RIP in our laboratory, which is inferred in [5]. RIP communicates with the computer via STM32F4 board as shown in Fig. 10. The two-variable static PID and PID with PID-adaptive combinations are implemented in the Matlab/Simulink environment. control block diagram is shown in Fig. 11.

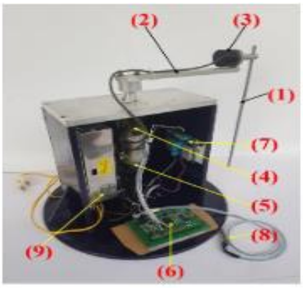

Fig. 10. Experimental system

(1)-pendulum
(2)-arm
(3)-encoder that measures angle of pendulum
(4)-motor
(5)-encoder that measures angle of motor (angle of arm)
(6)- STM32F4 board
(7)- H-bridge
(8)- cable that connects control board and PC

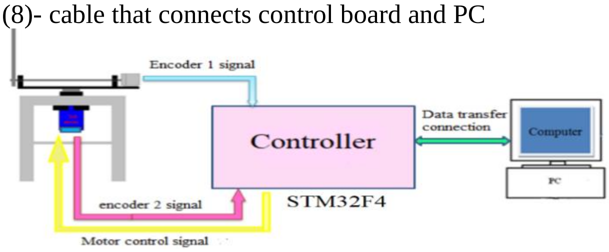

Fig. 11. RIP control diagram

PID-adaptive controller is applied only to control the pendulum angle. The arm angle is controlled by a static PID controller. The application of PID-adaptive to both the arm angle and the pendulum angle affects the compromise of two sets of PID-adaptive. From there, system destabilizes. Therefore, PID-adaptive will be applied to pendulum angle controller, static PID is applied to arm angle controller, final control signal is result of compromise of these above two controllers. The two-variable PID control algorithm flowchart incorporates the neural structure shown in Fig. 12.

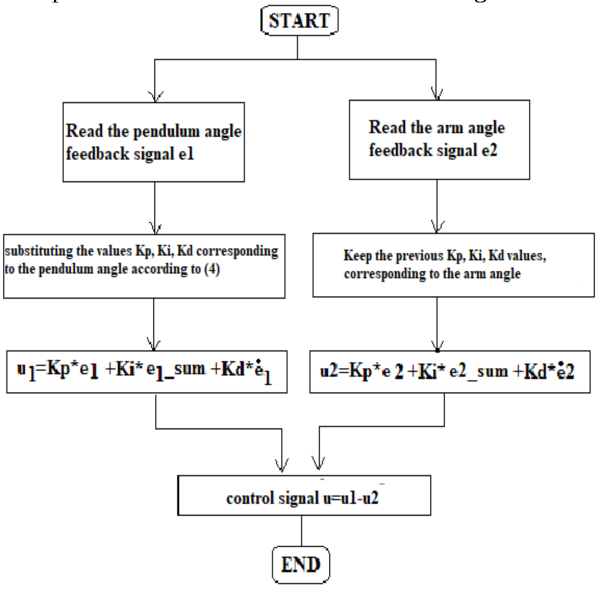

Fig. 12. Two-variable PID control algorithm flowchart incorporating neural structure

In Fig. 13, system is driven by a static two-variable PID set whose angular value deviates α fluctuates between [-3^0,2^0]. When RIP is controlled by PID-adaptive controller, vibration of pendulum was reduced from about [-3^0,2^0] to about [-2^0,1^0] as shown in Fig. 14. Process of changing parameters K_P, K_I, K_D shown in Fig. 15. Initially selected PID controller parameter value is $K_P = 11$, $K_I = 0.1$, $K_D = 11$ respectively. After a period of 35 seconds, PID parameters have reached a stable value $K_P = 12.5$, $K_I = 2.2, K_D = 10.3$

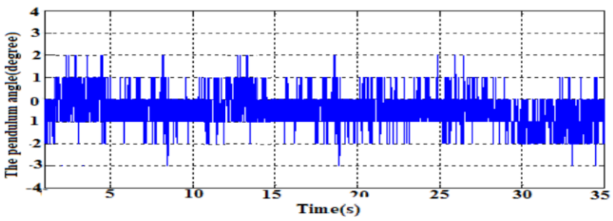

Fig. 13. Real system response to static two-variable PID controller

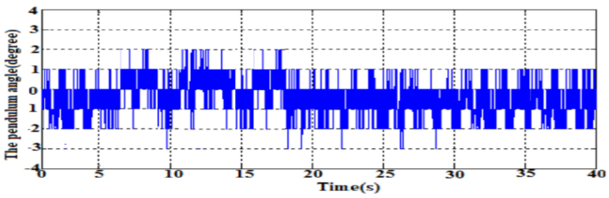

Fig. 14. Real system response to a PID controller with a neuron combination

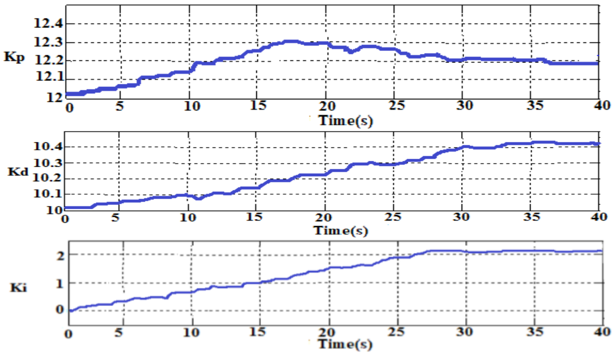

Fig. 15. The process of changeness of Kp, Ki, Kd

6. Conclusion

In this study, we presented a two-variable PID controller that incorporates a neural structure to tunning the value K_P, K_I, K_D online (PID-adaptive) for RIP. Simulation and experimental results showed that the integration of the neural structure to adjust the PID control parameter value online makes the system fluctuate less and stabilized over time. The value K_P, K_I, K_D also gradually changes in the optimal direction so that the system reaches a stable value.

Acknowledgement

We give thanks to PhD. Van-Dong-Hai Nguyen (HCMUTE) for supporting ideas and equipments for this research.

7. References

[1] Nath V., Mitra R.: "Swing-up and control of Rotary Inverted Pendulum using pole placement with integrator", Recent Advances in Engineering and Computational Sciences (RAECS), Chandigarh, India, 2014, pp. 1-5, 2014.

[2] Ionete C.: "LQG/LTR Controller Design for Rotary Inverted Pendulum Quanser Real-Time Experiment", Proceeding of the International Symposium on System Theory, Craiova, Romania, pp, 55-60, 2003.

[3] M A.L., Kunjumuhammed A. et al.: "Stabilization of Rotary Inverted Pendulum using PID Controller", 8th International Conference on Smart Computing and Communications (ICSCC), IEEE, pp. 376-380, India, 2021.

[4] Thành T.D.C.: "Mô phỏng Bộ điều khiển neuron với luật học hệ số học thích nghi và phương pháp xung lượng", Tạp chí phát triển KH&CN, tập 11, số 3, trang 69, 2008.

[5] Vo M.T. et al.: "Back-stepping control for rotary inverted pendulum", Journal of Technical Education Science, (59), 93–101, 2020.