DESIGN AND MANUFACTURE A ROBOT ARM CONTROL SYSTEM FOR EDUCATION AND TRAINING PURPOSES USING ARDUINO UNO

Nguyen Hoai Nam*, Nguyen Quang Minh
College of Technology - TNU

SUMMARY

Our target is to manufacture a robot arm control system, which can be utilized for experimental purposes. Firstly, a robot arm is designed and fabricated in the laboratory. It consists of a link, a dc motor and a gear and an encoder attached to the shaft of the motor. Next, an Arduino UNO is connected to a computer via a COM port. Then, a PID controller is designed and implemented using either Matlab/Simulink or the Arduino UNO. Finally, the robot arm is controlled in real time mode. The result shows that the control system is stable and the robot arm position tracks the target very closely. Other advanced controllers, such as fuzzy, neural network, nonlinear and even adaptive controllers, can also be applied to control the robot arm. The advantage of this control system is quite cheap in comparison with the similar ones provided by companies. In addition, the quality of the system is good enough to be accepted for experiment. Furthermore, it is very flexible to use other controllers.

Key words: *PID control, Arduino, robotarm, microcontroller*

INTRODUCTION

Our aim is to build an experimental model of control system, which can be used to test any control algorithms. The plant should be typical in area of nonlinear and dynamic systems. Thus, an one link robot arm is the simplest case to study not only in our interested area but also in robotics. For the control algorithm, we use the digital version of classical PID controller. Thus, we choose Arduino Uno as a hardware to program the algorithm.

It is not difficult to design and implement a PID controller but it provides very good performances such as stability, fast setting time and less overshoot. The PID controller was first formulated by Nicholas Minorsky in 1922 [1,2]. Ziegler and Nichols proposed and examined a tuning method for PID controllers [3] for the first time. There are also other methods of tuning PID controllers. Parameters of PID controllers are calculated by using a fuzzy and neural network [4]. An optimal tuning of PID controllers method for a class of first order and time delay systems [5] is proposed. [6] The design of PID controllers is developed for stable or unstable linear systems of arbitrary order with possible non-minimum-phase and irrational characteristics and including significant time delay. A method of obtaining PID parameters [7] is proposed for general process models by approximating the feedback form of an IMC controller with a Maclaurin series in the Laplace domain. A PID controller tuning method [8] is proposed. This method is based on the fitting of the process frequency response to a particular second-order plus dead time structure. A review and new results of PID tuning rules [9] for second order plus dead time systems are provided. A tuning strategy [10] for robust PID controllers satisfying multiple H∞ specifications is proposed. A design of an optimal disturbance rejection PID controller [11] is presented by using GAs.

It is estimated that [12] the share taken by PID controllers is more than 90%. A modern overview of functionalities and tuning methods in patents, software packages and commercial hardware modules [13] is offered. FPGA based PID controller [14] is implemented for a DC motor speed control. Design and Implementation of FPGA-based

* *Tel: 0917 987683, Email: hoainam@tnut.edu.vn*

PID Controllers [15] is proposed. An Arduino based PID controller [16] is implemented for the control of ionic polymer metalcomposite actuator. An Arduino based neural network controller [17] for the speed of a DC motor is implemented. In fact, there are many more works related to microprocessor usage, but we just focus on some most related ones.

The one link robot arm is the simplest one in the area of robotics, but it presents all characteristics of a nonlinear dynamic systems, robots and manipulators. A PI controller [18] was applied to control the position of the robot arm. For our project, we design a digital PI controller and implement based on Arduino Uno. In the next section, the design and the making process of a robotarm is mentioned. In the section 3, a digital PI controller is offered, codes for the PI algorithm are given and the real time control is done. The final section provides some results and future work.

ROBOT ARM CONTROL SYSTEM

A real model of robot arm control system is designed and made at the laboratory as shown in Fig. 1, where (1) is an Arduino Uno, (2) is a H-Bridge, (3) is an external setpoint, (4) is a DC motor attached with gear and encoder, (5) is a frame and (6) is a robot link.

An Arduino Uno is used as a hardware for the digital PID controller. It is programmed to read the angle position of the link from the encoder via Pins 2 and 3, and then compute the control signal by using a PID algorithm. The signal is used to drive a PWM pin that controls the H-bridge. A PWM voltage, proportional to the control signal, from the H-bridge is applied to the dc motor. Since we use the microcontroller, other control algorithms can also be programmed to regulate the position of the robot arm. This provides a flexibility in using advanced control.

An advantage of Arduino is that a computer can be used to indirectly control the plant via an Arduino. Fortunately, Matlab/Simulink provides an IO library for Arduino. Thus, we can setup the Arduino and do programming on Simulink to execute any algorithm. Therefore, there are two options for using hardware: Arduino and computer/laptop. For this set of hardwares, it is also used to control other single input single output plants such as magnetic levitation systems, inverted pendulums, robotics and manipulators. In the following section, we discuss about design of digital PID controller and implementation on Arduino Uno.

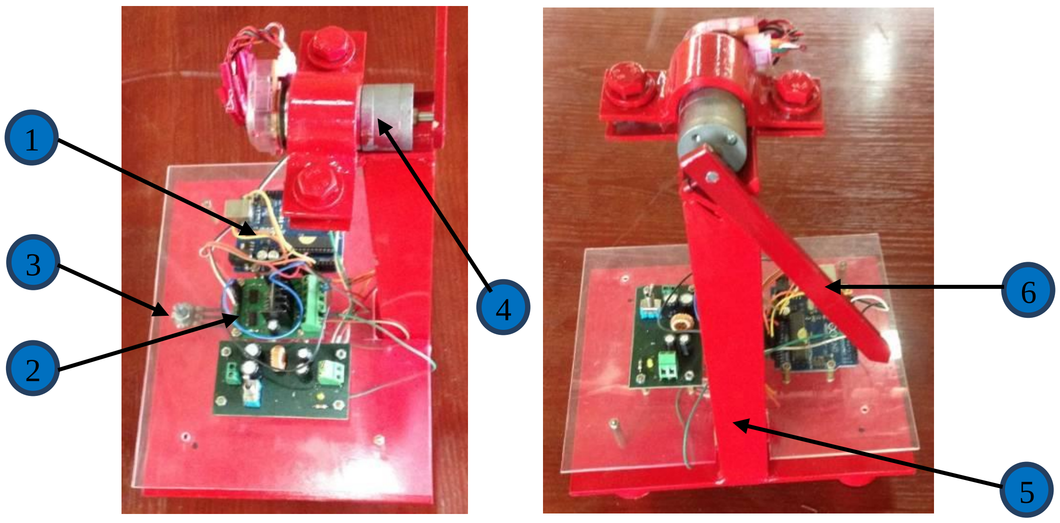

Figure 1. *A model of robot arm control system*

DESIGN, IMPLEMENTATION AND CONTROL

Currently, there are mainly two types of PID controllers: continuous-time PID (conventional) and discrete-time PID (modern). In this project, we use a discrete-time PID controller. To design it, there are several ways. It can firstly be designed in continuous-time domain using one of many methods such as Ziegler and Nichols [4]. But there must exists an exact mathematical model of the plant. Then, the contunuous-time PID is transformed into discreate-time domain by using some approximations for integral and derivative parts to obtain a discreate-time PID controller, called digital PID controller. It is actually a difference equation that can be programmed. For this type of controller, we need to determine a appropriate sampling time T_s before doing any approximation. The other way to design a digital PID is to do it in discreate-time domain. If there exist a precise mathematical model of the plant in discreate-time domain, then it is easier to do. Otherwise, trial and error method can only be used. Thus, we apply this method to our system because it is not easy to identify our self-made robotarm and also unneccessary for experimental purposes. The difference equation for the digital PID controller can be represented as the following form;

$$u_k = \left[k_p e_k + k_i(e_k + e_{k-1}) + k_d(e_k - e_{k-1})\right]/T_s \quad (1)$$

where e_k is the error between the setpoint and the plant output, u_k is the calculated control signal and the digital PID parameters proportional gain, integral gain, derivative gain and sampling time are k_p, k_i, k_d and T_s, respectively. They are determined by some experiments and their values are as follows

$$k_p = 0.067; \quad k_i = 0.007; \quad k_d = 0.0005$$
and $T_s = 0.05\ (\text{sec})$. (2)

Based on Eq. (1) and Eq. (2), the sketch of the digital PID controller for Arduino is written as follows,

Main program

```
#define CHA 2; #define CHB 3; #define
PWM_PIN 6; #define DIR_PIN 12
volatile int feedback = 0;float out_motor =
0;intref_point;intrefe;
void setup(){
pinMode(CHA, INPUT);pinMode(CHB,
INPUT);pinMode(PWM_PIN, OUTPUT);
pinMode(DIR_PIN,
OUTPUT);Serial.begin(9600);attachInterrupt(
0, readfeedback, RISING);}
void loop(){
Serial.println(map(feedback,0,9666,0,360));
Serial.print ("\t");
Serial.print(ref_point); Serial.print
("\t");read_ref();
out_motor =
out_PID(map(ref_point,0,360,0,9666),feedba
ck,0.067,0.007,0.0005);
if
(out_motor>0){digitalWrite(DIR_PIN,HIGH)
;analogWrite(PWM_PIN,abs(out_motor));}
if
(out_motor==0){digitalWrite(DIR_PIN,LOW
);digitalWrite(PWM_PIN,LOW);}
if
(out_motor<0){digitalWrite(DIR_PIN,LOW);
analogWrite(PWM_PIN,abs(out_motor));}
}//endloop
void readfeedback (){if (digitalRead(CHA)
&& !digitalRead(CHB)) {feedback++ ;}
if (digitalRead(CHA) &&digitalRead(CHB))
{feedback-- ;}} // end
voidread_ref(){refe=analogRead(A0);ref_poi
nt = map(refe,0,1022,-180,180);} // end
```

PID code

```
int
error=0;intlast_error=0;intPterm=0,Dterm=0,I
term=0;intintegrate_error=0;
float out_PID ( float set_point , float
feed_back , float Kp , float Kd , float Ki){
error = set_point - feed_back;Pterm =
Kp*error/0.05;integrate_error += error;
Iterm = Ki*integrate_error/0.05;Dterm = Kd*
(error-last_error)/0.05;last_error=error;
```

return constrain (Pterm+Iterm+Dterm,-255,255);delay(20);}

This sketch is loaded up to the Arduino via the COM 3 to run the control system in real time mode. The setpoint (channel 2), the position (channel 1) of the link and the error are shown in Fig. 2. It can be concluded that the control system is stable, it runs very smoothly and the position of the link tracks closely the target signal. The steady state error is very small, the setting time is short and the overshoot is in the permitted range.

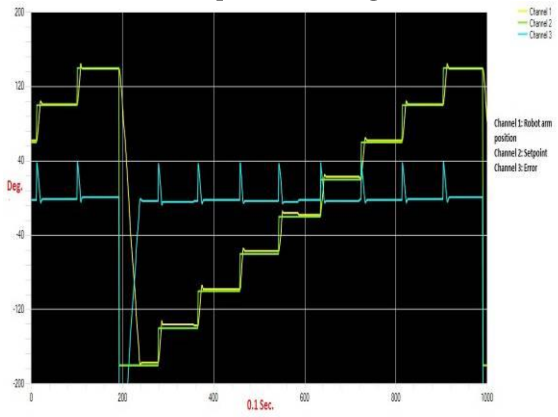

Figure 2. *Real-time position of the link, the setpoint and the error*

CONCLUSIONS

In summary, an experimental model of robot arm control system is built at the laboratory and an Arduino based PID controller is implemented. This robot arm control system based on Arduino/Computer can be used for experimental purposes such as studies on robotics, nonlinear dynamic systems, microcontroller, C/C++ programming, real-time toolbox of Matlab, position/speed sensors, H-bridges, DC motor control as well as research, for example, test new control algorithms and try advanced control algorithms like fuzzy, neural networks and adaptive control. In addition, this control system can be applied to control other SISO plants like magnetic levitation systems, inverted pendulums and some MIMO plants like manipulators.

Future works will focus on improving the performance of the system by using advanced controllers, adding an current control loop and doing plant identification.

Acknowledgments

This project hardware was partly helped by the FIT's Center of Discovery and funded by the research project T2014-18.

REFERENCES

1. S. Bennett, "Nicholas Minorsky and the automatic steering of ships," IEEE Control systems magazine, Vol. 4, Iss. 4, 1984.
2. S. Bennett, "A brief history of automatic Control," IEEE Control Systems Magazine, IEEE, Vol. 16, Iss. 3, 1996.
3. J.G. Ziegler and N. B. Nichols, "Optimum settings for automatic controllers," Trans. ASME, Vol. 64, No. 8, 1942.
4. C. H. Lee and C. C. Teng, "Calculation of PID controller parameters by using a fuzzy neuralnetwork," ISA Transactions 42, p. 391–400, 2003.
5. S. Tavakoli and M. Tavakoli, "Optimal tuning of PID controllers for first order plus time delay models using dimensional analysis," The fourth Int. Conf. on Control andAutomation, Montreal, Canada, 2003.
6. Z. SHAFIEIT and A. T. SHENTON, "Frequency-domain Design of PID Controllers for stable and unstable systems with time delay," Automatica. Vol. 33, No. 12, 1997.
7. Y. Lee, S. Park, M. Lee and C. Brosilow, "PID Controller tuning for desired closed-loop responses for SI/SO systems," AICHE Journal, Vol. 44, No. 1, 1998.
8. Q. G. Wang, T. H. Lee, H. W. Fung, Q. Bi, and Y. Zhang, "PID Tuning for improved performance," IEEE Trans. On Control Systems Technology, Vol. 7, No. 4, July 1999.
9. R. C. Panda, C. C. Yu and H. P. Huang, "PID tuning rules for SOPDT systems: Review and some new results," ISA Transactions 43, p. 283–295 2004.
10. T. H. Kim, I. Maruta and T. Sugie, "Robust PID controller tuning based on the constrained particle swarm optimization," Automatica, August 2007.
11. R. A. Krohling and J. P. Rey, "Design of optimal disturbance rejection PID controllers using Genetic Algorithms," IEEE Trans. On Evolutionary Computation, Vol. 5, No. 1, February 2001.
12. V. KUMAR, B. C. NAKRA and A. P. MITTAL," A Review on Classical and Fuzzy PID Controllers," Int. Journal of Intelligent Control and Systems, Vol.16, No. 3, 2011.

13. K. H. Ang, G. Chong, and Y. Li, "PID Control System Analysis, Design, and Technology," IEEE Trans. On Control Systems Technology, Vol. 13, No. 4, July 2005.
14. F. H. Ali, M. M. Hussein and S. M. B. Ismael, "LabVIEW FPGA Implementation Of a PID Controller For D.C. Motor Speed Control," Iraq J. Electrical and Electronic Engineering, Iraq J. Electrical and Electronic Engineering, Vol. 6 No. 2, 2010.
15. Y. F. Chan, M. Moallem, and W. Wang, "Design and Implementation of Modular FPGA-based PID Controllers," IEEE Trans. Industrial Electronics, Vol 54, Iss. 4, Aug. 2007.
16. M. Rosly, Z. Samad and M. F. Shaari, "Feasibility Studies of Arduino Microcontroller Usage for IPMC Actuator Control," IEEE International Conference on Control System, Computing and Engineering, 28 - 30 November 2014, Penang, Malaysia.
17. N. Rai, B. Rai, "Neural Network based Closed loop Speed Control of DC Motor using Arduino Uno,"International Journal of Engineering Trends and Technology, Volume 4, Issue 2, 2013.
18. M. T. Soylemez, M. Gokasan, O. S. Bogosyan, "Position Control of a Single-Link Robot- Arm Using a Multi-Loop PI Controller", Proceedings of 2003 IEEE Conference on Control Applications.

TÓM TẮT
THIẾT KẾ VÀ CHẾ TẠO HỆ THỐNG ĐIỀU KHIỂN CÁNH TAY MÁY VỚI MỤC ĐÍCH GIÁO DỤC VÀ ĐÀO TẠO SỬ DỤNG ARDUINO UNO

Nguyễn Hoài Nam[*], Nguyễn Quang Minh
Trường Đại học Kỹ thuật Công nghiệp – ĐH Thái Nguyên

Mục tiêu của chúng tôi là chế tạo một hệ thống điều khiển cánh tay máy, hệ thống này có thể được dùng cho các mục đích thí nghiệm. Đầu tiên, một cánh tay máy được thiết kế và lắp ráp trong phòng thí nghiệm. Hệ thống bao gồm một cánh tay, một động cơ một chiều được gắn hộp giảm tốc và thiết bị đo góc. Tiếp theo, một cạc Arduino Uno được nối với máy tính qua cổng COM. Sau đó, một bộ điều khiển PID được thiết kế và được thực hiện sử dụng Matlab/Simulink hoặc Arduino Uno. Cuối cùng, cánh tay máy được điều khiển thời gian thực. Kết quả cho thấy, hệ thống ổn đinh và vị trí của cánh tay bám rất sát tín hiệu đặt. Các bộ điều khiển nâng cao khác như mờ, nơ-ron, phi tuyến và thậm chí là thích nghi cũng có thể được áp dụng để điều khiển cánh tay máy. Ưu điểm của hệ thống điều khiển này là khá rẻ so với các hệ thống tương đương cung cấp bởi các công ty. Hơn nữa, chất lượng của hệ thống đủ cao để được chấp nhận cho thí nghiệm. Thêm vào đó, hệ thống rất linh hoạt để sử dụng các bộ điều khiển khác.
Từ khóa: *điều khiển PID, Arduino, cánh tay máy, vi điều khiển*

Ngày nhận bài:20/6/2015; Ngày phản biện:06/7/2015; Ngày duyệt đăng: 30/7/2015
Phản biện khoa học: *PGS.TS Nguyễn Duy Cương - Trường Đại học Kỹ thuật Công nghiệp - ĐHTN*

[*] *Tel: 0917 987683, Email: hoainam@tnut.edu.vn*