VOL. 10, NO. 17, SEPTEMBER 2015 ISSN 1819-6608
ARPN Journal of Engineering and Applied Sciences

www.arpnjournals.com

ARDUINO BASED ELECTRO-MECHANICAL THROTTLE CONTROLLER FOR AUTOMOTIVE APPLICATIONS

Bambang Supriyo[1, 2], Kamarul Baharin Tawi[1], Mohd Salman Che Kob[1] and Izhari Izmi Mazali[1]
[1]Faculty of Mechanical Engineering, UniversitiTeknologi Malaysia, Johor Bahru, Johor, Malaysia
[2]Department of Electronic Engineering, PoliteknikNegeri Semaran, Central Java, Indonesia
E-Mail: bambang@fkm.utm.my

ABSTRACT

This paper introduces an electro-mechanical throttle actuator and implements its real time Proportional-Integral-Derivative (PID) based position controllers using an Arduino Uno microcontroller and Matlab/Simulink® software. The Arduino Uno acts as an inexpensive USB based Data Acquisition System. It is controlled by a Matlab/Simulink program in a host computer to perform PID based throttle controller tasks. The throttle actuator is based on a linear actuation DC motor which is directly connected to the engine throttle lever using a metal cable. Initial PID parameters are obtained experimentally using combinations of relay feedback method and Ziegler Nicholl formula. The performance of throttle actuator controller is assessed in terms of percent overshoot, settling time and steady state error. The results show that the Proportional-Derivative (PD) and Proportional-Derivative-Plus-Conditional-Integral (PDPCI) controllers have adequately improved Proportional (P) controller performance, especially in terms of overshoots in which the smallest overshoot of 1 percent is achieved by PDPCI controller.

Keywords: throttle, electro-mechanical, PID, arduino uno.

INTRODUCTION

In automotive application, throttle valve controls the amount of air and fuel flowing into the engine's combustion chamber to produce a relatively constant air/fuel ratio value, and thereby regulates the amount of power generated by the engine [1-2].On an engine equipped with a conventional carburetor, the accelerator pedal is mechanically connected to the throttle valve by means of metal cable. As the pedal is depressed, the throttle valve opens allowing more air and fuel to flow into the combustion chamber, hence increasing engine power accordingly. Unfortunately, this conventional carburetor results in less engine efficiency [3]. Advancement in electronics technology has resulted in a drive-by wire technology in automotive applications [4-5]. Based on this technology, the throttle and accelerator pedal become two independently separate systems. In this case, the throttle valve opening is not mechanically linked to the acceleration pedal, but it is electronically linked. Throttle valve is actuated by a DC servomotor and its angular valve position is detected by a rotary position sensor, while the angular position of the accelerator pedal is also sensed using other rotary position sensor. The Engine Control Unit (ECU) reads the set point obtained from the current position of the acceleration pedal and then controls the DC motor to produce an exact position of the throttle valve guided bythethrottle valve position sensor. The desired throttle opening is calculated by ECU based on the required air flow determined by the engine speed and torque [6]. This drive by wire technology has resulted in drivability, fuel economy and emission improvements [7-8].In this research, a throttle actuator system is developed based on a linear actuation DC motor as its actuator in which its plunger is directly connected to the throttle lever to open or close the throttle valve. Inside the linear actuator, there is a position sensor which detects the stroke distance of the actuator. A PC based controller for an external electromechanical throttle actuator (EMTA) is implemented in real time using Matlab/Simulink and Arduino Uno as a USB based data acquisition system. Arduino Uno is low-cost microcontroller board based on ATMega328 system [9], as shown in Figure-1. It has 14 configurable digital input/output pins, in which 6 of them can be used as pulse width modulation (PWM) outputs and 6 analog input pins. In this research, the Arduino Uno is pre-programmed and configured as a data acquisition system which communicates with Matlab/Simulink via USB port [10]. Recently, there have been quite numbers of control applications involving Arduino microcontroller system and Matlab/Simulink, such as mobile robot [11], water level control system [12] and solar tracker system [13].

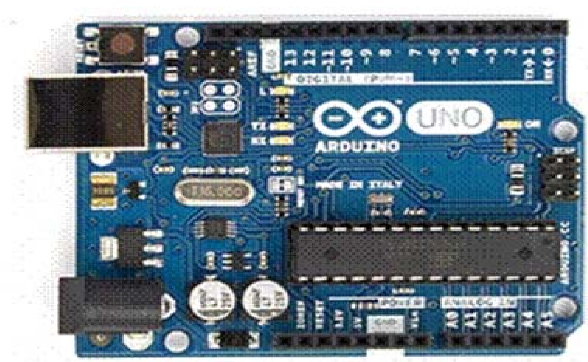

Figure-1. Arduino Uno Board [7].

Proportional Integral Derivative (PID) based controllers, such as Proportional Derivative (PD) and Proportional Derivative Plus Conditional Integral (PDPCI) [14] are implemented using Matlab/Simulink to control the

VOL. 10, NO. 17, SEPTEMBER 2015 ISSN 1819-6608
ARPN Journal of Engineering and Applied Sciences

angular position of the throttle valve. A combination of Åström-Hägglund Relay Feedback experiment and Ziegler-Nichols formula is utilized to obtain initial PID parameters [15]. The throttle controller performances are assessed based on the percent overshoot, settling time and steady state error. This throttle actuator system can be used in an engine test requiring various throttle opening positions in which its measurement results can be lodged in the computer.

ELECTRO-MECHANICAL THROTTLE CONTROLLER

System description

An electro-mechanical throttle actuator (EMTA) system, as shown in Figure-2, mainly consists of a linear actuation DC motor with built in position sensor to detect actuator's stroke position, throttle system, computer with Matlab/Simulink, Arduino Uno based data acquisition system, actuator driver, and DC power supply. The plunger of the linear actuator is directly linked to the throttle lever using a metal cable. The throttle angle is indirectly sensed using built in position sensor of the linear actuator. This angle range is between 0 and 90 degrees. The output voltage of this position sensor is fed to the analog input pin (A0) of Arduino Uno. An actuator driver is used as an interfacing unit between Arduino Uno and the linear actuator. This actuator driver is controlled by one digital output pin (Pin2) of Arduino Uno for changing the direction of the plunger, either forward or reversed and one PWM output pin (Pin3) of Arduino Uno for regulating the speed of the plunger. The Arduino Uno communicates with Matlab/Simulink program via USB port.

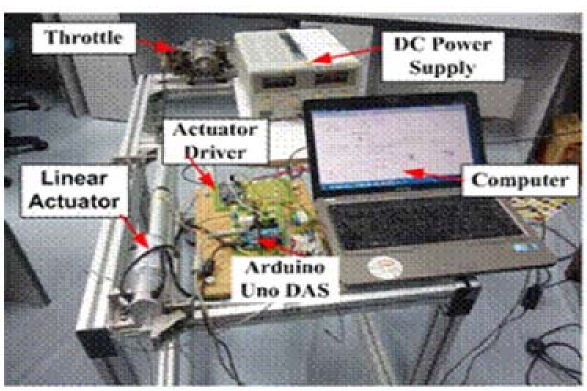

Figure-2. Pulley clamping force test rig.

Initial PID parameters

Relay feedback experimental methods have been extensively used in tuning PID controllers automatically. The relay feedback experimental method has been suggested by Åström and Hägglund (1984) to generate a bounded continuous oscillation in which the estimated critical information of oscillation period (T_c), oscillation amplitude (a) and relay voltage (h) can be extracted to obtain the critical gain (K_c) of a PID controller by using Eqn. (1). Based on both critical period and critical gain, the initial PID parameters of proportional (K_p), integral (K_i) and derivative (K_d) gains can be determined using Eqns. (2), (3) and Ziegler-Nichols formula as given in Table-1.

$$K_c = 4h/\pi a \quad (1)$$

$$K_i = K_p/T_i. \quad (2)$$

$$K_d = K_p T_d. \quad (3)$$

Table-1. Ziegler-Nichols formula.

Controller	K_p	T_i	T_d
PID	0.60 K_c	0.50 T_c	0.125 T_c

PDPCI controller

PDPCI controller is a PID based controller consisting of PD controller and integral controller in which the integral part will be activated if the system error is approaching to zero [15]. In this case, a predetermined conditional value for integral activation is compared with the absolute value of the system error. If the system error is bigger than the conditional value, the PDPCI will work as a standard PD controller; else the PDPCI will operate as a standard PID controller with its integral part gradually eliminating the steady state error.

RESULTS AND DISCUSSION

The results of relay feedback experiment are shown in Figure-3 and Figure-4. Based on the values of amplitudes in Figure-3 and relay output in Figure-4, the critical gains (Kc) can be calculated using Equation (1). By taking the average values of period in Figure-4, the critical period (Tc) can be obtained. The initial PID parameters are computed using Ziegler-Nichols formula in Table-1, Equations (2) and (3). The resulted initial PID parameters for opening and closing the throttle valve are presented in Table-2, while the controller performances are shown in Figure-5 to Figure-9, for step input of 10, 30, 50, 70 and 90 degrees. Each step input starts at 0.1 second.

Almost all controllers produce overshoots, except for set point of 90 degrees, since this point is the maximum limit for the throttle valve angle. For setpoints of 10, 30, 50 and 70 degrees, P controller produces the highest overshoots of 13.0, 6.0, 4.0 and 3.0 percents respectively, while PD controller results in overshoots of 5.0, 2.0, 2.0 and 1.4 percents, respectively and PDPCI controller results in overshoots of 3.0, 1.0, 1.0 and 1.4 percents, respectively. So, the PDPCI produces the lowest overshoot of 1 percent for set point of 30 and 50 degrees. In terms of settling time, P, PD and PDPCI controllers produce the same settling time of 0.15, 0.3, 0.45, 0.6 and 0.83 seconds for set points of 10, 30, 50, 70 and 90 degrees, respectively. It shows that the settling time is linearly proportional to the set point. From Figure-9, it can be seen that the fastest time required for the throttle valve

ARPN Journal of Engineering and Applied Sciences

to open from 0 degree to 90 degrees is about 0.83 seconds. In terms of steady state error, all controllers successfully reduce the steady state error to about 0 degrees. Overall, for all set points, the PDPCI and PD controllers have similar performances in terms of settling time, but PDPCI is slightly better in terms of overshoot and steady state error reductions. The biggest overshoot and steady state error for all controllers occur when the set point is 10 degrees. It is likely that small set points require less proportional gain to reduce the overshoot and steady state errors. In this case, autotuning based on adaptive based controllers such as Fuzzy controller, Neural Network, etc, can be proposed to dynamically tune the PID parameters.

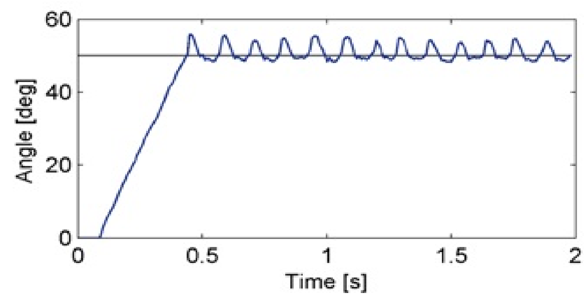

Figure-3. Output oscillation.

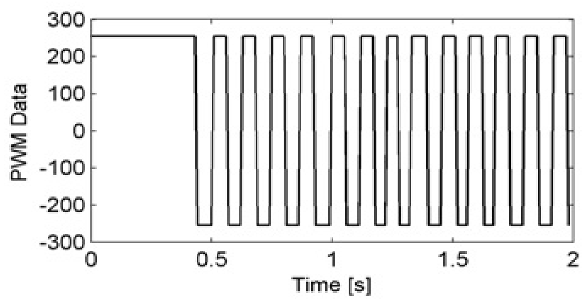

Figure-4. Relay output.

Table-2. PID parameters.

	K_p	K_i	K_d
Open	117.5	30	0.9
Close	40.5	10	0.3

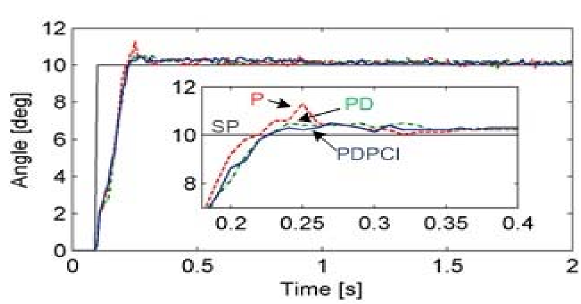

Figure-5. Output response (set point: 10 degrees).

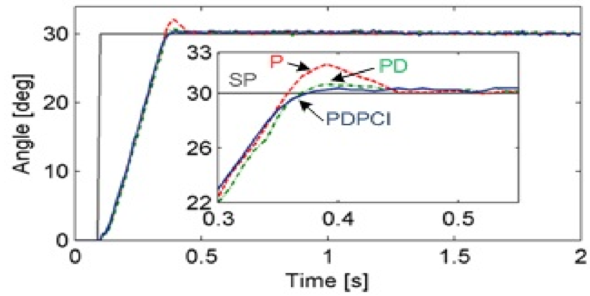

Figure-6. Output response (set point: 30 degrees).

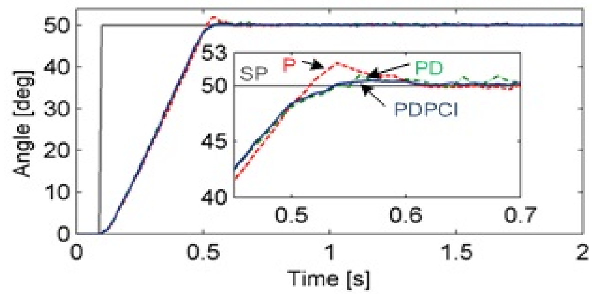

Figure-7. Output response (set point: 50 degrees).

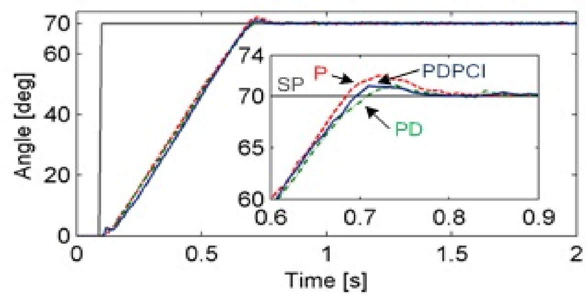

Figure-8. Output response (set point: 70 degrees).

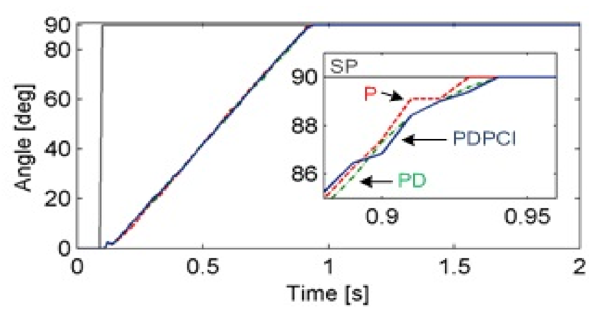

Figure-9. Output response (set point: 90 degrees).

CONCLUSIONS

By using initial PID parameters obtained from relay feedback experimental method, PID based throttle controllers have shown good performance. It proves that the relay feedback method can be an acceptable and effective method of providing initial PID parameters for

PID based control algorithms used in this electro-mechanical throttle controller system. The step inputs of 10, 30, 50, 70 and 90 degrees are used to test the system response performances of P, PD and PDPCI controllers. Overall, the PDPCI controller has adequately improved the P and PD performance in terms of settling time and steady state error, although all of the controllers have performed much similar results with small tolerable differences between them.

ACKNOWLEDGEMENTS

The authors would like to express their appreciation to Ministry of Higher Education (MOHE) and Universiti Teknologi Malaysia (UTM) for continuous supports in the research work. This work was financially supported by UTM Tier-2 GUP-2013 Vot Number 09J15.

REFERENCES

[1] Noor M. M., Kadirgama K., Devarajan R., Rejab M. R. M., Zuki N. M. and Yusaf T. 2008. Design of a High Pressure Compressed Natural Gas Mixer for 1500cc Engine. In: National Conference on Design and Concurrent Engineering. pp. 435-438.

[2] Ramasamy D., Bakar R. A., Rahim M. F. and Noor M. M. 2008. Comparative Evaluation of a Two Stroke Compressed Natural Gas Mixer Design using Simulation and Experimental Techniques. In: Asia Power And Energy Systems. pp. 359-362.

[3] M. Corno, M. Tanelli, S. S. Savaresi and L. Fabbri. 2011. Design and validation of a gain-scheduled controller for the electronic throttle body in ride-by-wire racing motorcycles. IEEE Transaction on Control Systems Technology. Vol. 19, pp.18-30.

[4] Corno M., Tanelli M., Savaresi S.S., Fabbri L. and Nardo L. 2008. Electronic throttle control for ride-by-wire in sport motorcycles. In: 17th IEEE International Conference on Control Applications Part of 2008 IEEE Multi-conference on Systems and Control. pp. 233-238.

[5] N. Jamil, A. R. Yusoff, M. H. Mansor. 2012. Literature review on electromagnetic actuator force generation for dynamic modal testing applications. Journal of Mechanical Engineering and Sciences. Vol. 3, pp. 311-319.

[6] Y. Pan, U. Ozguner and O. H. Dagci. 2008. Variable structure control of electronic throttle valve. IEEE Transaction on Industrial Electronics. Vol. 11, No. 55, pp. 3899-3907.

[7] T. Aono and T. Kotawari. 2006. Throttle control algorithm for improving engine response based on air-intake model and throttle response model. IEEE Transaction on Industrial Electronics. Vol. 53, No.3, pp. 915-922.

[8] M. Vasak, M. Baotic, I. Petrovic, and N. Peric. 2007. Hybrid theory based time optimal control of an electronic throttle. IEEE Transaction on Industrial Electronics. Vol. 54, No. 3, pp. 1483-1494.

[9] Information on http://en.wikipedia.org/wiki/Arduino. Visit on 22 August 2013.

[10] Manoiu-Olaru S. and Nitulescu M.. 2011. Hexapod robot. Virtual models for preliminary studies. In: 15th international conference on system theory, Control and Computing (ICSTCC). pp. 1-6.

[11] Gartseev I. B., Lee L. F. and Krovi V. 2011. A low-cost real-time mobile robot platform (ArEduBot) to support project-based learningin robotics & mechatronics. In: Proceedings of 2nd International Conference on Robotics in Education (RiE 2011). pp. 117-124.

[12] Krivić S., Hujdur M., Mrzić A. and Konjicija S. 2012. Design and implementation of fuzzy controller on embedded computer for water level control. In: MIPRO Proceedings of the 35th International Convention. pp. 1747-1751.

[13] D. Bawa and C. Y. Patil. 2013. Fuzzy control based solar tracker using Arduino Uno. International Journal of Engineering and Innovative Technology. Vol. 2, No. 12, pp. 179-187.

[14] B. Supriyo, K. B. Tawi and H. Jamaluddin. 2013. Experimental study of an electromechanical CVT ratio controller system. International Journal of Automotive Technology. Vol. 14, No. 2, pp. 313-323.

[15] K. J. Åström and T. Hägglund. 1984. Automatic tuning of simple regulators with specifications on phase and amplitude margins. Automatica. Vol. 20, No. 5, pp. 645–651.