

# Real Time DC Motor Position Control Using PID Controller in LabVIEW

Mustafa Saad<sup>1\*</sup>, ALHussein Amhedb<sup>2</sup>, Mohammed Al Sharqawi<sup>3</sup>

<sup>1, 2, 3</sup> Department of Control Engineering, College of Electronic Technology, Bani Walid, Libya  
Email: <sup>1</sup> mustafasaad9@yahoo.com

**Abstract**— Direct current (DC) motors are the most used motors in control engineering application due to their simplicity of construction, easy to control and excellent performance. These motors should be well controlled to perform the required task. The functional application of DC motor is focused on this research by using LabVIEW for position control system. A closed loop real-time control system with an added 298 encoder coupled to the motor shaft is used in this control system to provide feedback position signals to the Proportional Integral Derivative (PID) controller. The PID controls the position of the DC motor at the desired position with a minimum error. The PID controller was implemented in LabVIEW software which sends the control signal to the real time DC motor through the Arduino board. In addition, LabVIEW software was developed to show the output response of motor position versus time to easily observe the performance of the system. The PID controller gains were obtained based on trial and error method. The system under these controller parameters has been tested at different positions of tracking signal and for disturbance rejection. Finally, the results showed that the designed controller had good performance characteristics where the desired position of the motor was maintained.

**Keywords**—DC motor, position control, PID controller, Arduino, LabVIEW

## I. INTRODUCTION

DC motor is one of the actuators that is widely used as a positioning system. It is used for many mechatronic system applications in world industries such as robots, precision positioning machines, or industrial applications. Accurate response in a predictable and repeatable manner is used for position control in mechatronic devices. For example, component laying must be right on a PCB board before the soldering process is carried out in a pick-and-place machine for printed circuit board production. Several motors for positioning the 2D or 3D robotic arm are in control of the robot manipulator [1,2,3]. Another example, a motor by means of control is applied to open the valve to drain the fluid. In various industrial applications, DC motors are often used in a variety of responses such as to follow a predetermined speed or position trajectory under variable load [4]. The position of the DC motor is controlled using various strategies and strategies. Some applications of DC motor speed and position controlled by previous researchers such as real-time controlled DC motor speed, control in learning objectives using operational amplifiers as analog PID controllers [5], digital control for DC motor speed and position [6], Position Based on the incremental encoder and speed indicator modeling and simulation [7], digital control of the servo system in the MS-150 modular position [8], and

real-time motor position control with a PID controller using LabVIEW [9].

DC motors are used in satellite antenna, robotic arm and crane. These motors are subjected to wear due to hard condition where it should give an accurate position as well as fast response with zero steady state error. To overcome these problems, a designed controller is needed to control the position of the DC motor perfectly. This is where the suggested PID controller comes in.

The main aim of this research is to design a PID controller to control the position of the real time DC motor, implement the controller in LabVIEW and interface the LabVIEW with the Arduino to obtain the position response of the real time DC motor as well as analyze the performance of the output responses.

## II. MODEL IMPLEMENTATION

In this paper, the structure of a feedback control system was constructed as illustrated in the block diagram of Fig. 1. From the figure, it can be seen that there is a sensor input that is fed back to produce an error value. This value is processed using Labview.

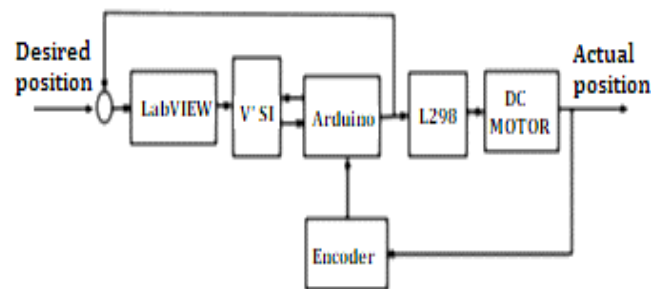


Fig. 1. Block diagram of overall system

The block diagram in the Figure shows that in this case it consists of a DC motor and an encoder sensor mounted on the motor shaft to measure the angular position. The feedback proportional integral control algorithm is derived from a simple control algorithm which is the basic algorithm used to control linear time invariance systems. The error is corrected and the transient response between the actual position and the desired position is increased by calculating the corrective actions that can change the process during operation, control and implementation of PID controls in LabVIEW. The virtual instrument software architecture (VISA) method is used as the interface between LabVIEW and Arduino in this paper

where this sub-function sends and receives data between LabVIEW and Arduino. Sending and receiving of sample data is used as a standard procedure for VISA and its operation. The time of receipt and delivery should be considered time delays. This paper presents a stable system with a set delay time of 6 msec, where each change in time delay causes instability in the system. This sub-function contains several important functions that send and receive data to LabVIEW and Arduino. The measured position of the encoder is received by arduino uno with the 8586 microcontroller and sent to LabVIEW.

#### A. Hardware Implementations

This section explains the components that had been used including main circuit, DC motor, DC motor driver, and incremental encoder.

The DC motor in this research is model RS-336SH-321 as shown in Fig. 2. This motor is paired with sensor which an incremental encoder to provide position feedback.



Fig. 2. Real DC motor model RS-336SH-321

A voltage of 6 to 24 volts DC is used for a DC motor to work and a current of less than 3 Amps is used to drive the rotor. This current is usually called the dead current because the current is drawn to the maximum by the motor. The motor can work with a maximum voltage of 24 Volt Dc but with a value of 12 Volt DC the motor can work for a short time to get a higher power level if the amplifier used to run the motor can provide the required power.

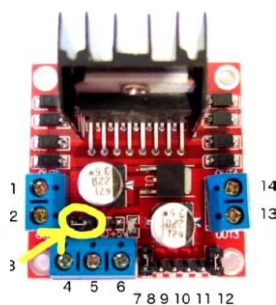


Fig. 3. L298N H-bridge pinouts

The L298N DC motor driver is a dual full-bridge driver designed to accept TTL logic levels capable of being used at high voltages and currents used to drive inductive loads such as relays, solenoids, DC and stepper motors. H-Bridge is used by these drivers for work. current flows in both directions using an H-bridge circuit [10]. The H-bridge L298N module can be used for motors with a voltage between 5 and 35V DC as shown in Figure 3. From the picture it can be seen that there is a 5V regulator, so the 5V voltage from the board can be used if your supply voltage reaches 12V.

From the figure, it can be seen that the module has a positive power supply at pin 4 and negative / GND at pin 5. The 12V jumper on the yellow circle pin 3 in Fig. 3 can be removed when the supply voltage is up to 12V and in this module a 5V voltage is provided at pin 6. The 5V voltage from the Arduino can be given to power it from the motor power supply module. The GND voltage on the Arduino is connected to the module.

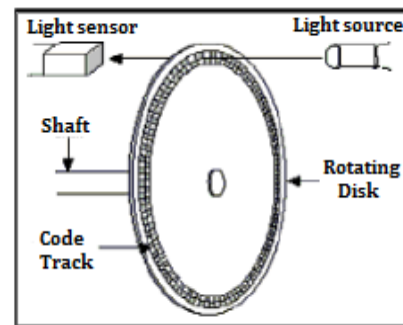


Fig. 4. Simple diagram of incremental encoder

Movement or rotating position is converted into a series of electronic pulses using an encoder sensor. Measuring sensors for rotary motion and for linear motion are used with rotary encoder sensors. Mechanical measurement standards such as lead screws can be used in conjunction with sensors. The incremental or absolute rotary motion is converted into an electrical signal. Low cost effective feedback devices have both [11].

Incremental encoder sensor referred to as optical encoder sensor is the most popular type of encoder because it consists of a rotating disk, a light source, and a photo detector (light sensor). The disc on the sensor has an opaque and transparent sector pattern used for coding mounted on a rotating motor shaft shown in Figure 4. This sector pattern can block the light emitted onto the photo detector resulting in a digital signal or pulse output [12].



Fig. 5. Main circuit components of this research

The output of the motor in the form of rotation, speed and position of the motor is combined to the control circuit. Problematic and non-conforming motor rotation can be detected by the feedback system. The problematic motor rotation is fixed with a control circuit until the motor finally reaches the desired point. Angle position information based on measurement in the form of signal is often needed in control system design. The main circuit, motor drive and motor with an incremental encoder are three important

components that are presented in this paper. The components of this research system circuit are shown in Figure 5.

### B. PID Controller Design

The process set point can be achieved and maintained by using one of the most widely used controls, namely PID control [13] - [18]. PID control applications can vary with the approach in each case remaining the same. The PID controller equation is expressed in ideal parallel form as

$$u(t) = K_p * e(t) + K_i \int_0^t e(t)de(t) + K_d \frac{de(t)}{dt} \quad (1)$$

where  $K_p$  is a constant variable for proportional gain,  $K_i$  is a constant variable for integral strengthening,  $K_d$  is a constant variable for derivative gain,  $e(t)$  is a variable for the

error value and  $u(t)$  is a variable for control signals.  $K_p$ ,  $K_i$  and  $K_d$  are control constant variables which have non-negative values.

Proportional control has a variable  $K_p$  is used as an amplification which can reduce the rise time but can generate and increase the error [19] - [22]. The integral control has variable  $K_i$  used to eliminate steady-state errors but can make transient responses worse. Derivative control has the  $K_d$  variable used to increase system stability, reduce overshoot, and increase transient response. PID blocks are used in LabVIEW to implement PID control and simulate it. The main elements of each block diagram are shown in Fig. 6. The controller parameters were adjusted using the experimental method of PID parameter improvement.

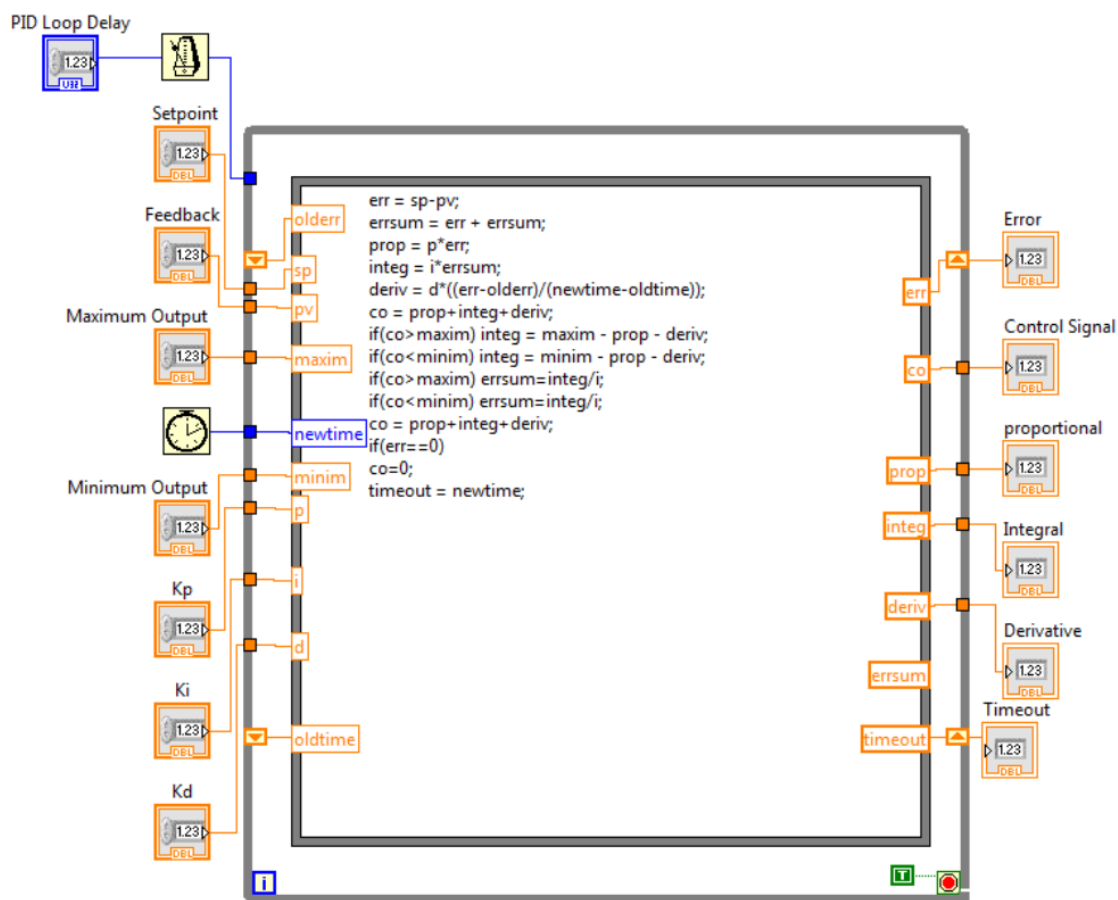


Fig. 6. The designed PID controller in LabVIEW.

The figure shows a digital PID control flow that has several inputs and outputs, which are made in the form of a script code (sub code). The value of the input signal, control variable, sensor signal and PWM signal has a value ranging from -255 to 255. The output is the sum of the feedback signal with the setpoint is an error which is processed into a control signal and used for PID control. The time delay is set to zero in this loop because it is already in the main loop.

### C. System Wiring

Firstly, the Dc motor was connected to A and B connections on the L298N module. In addition, power supply was connected with the positive power supply sign on the module and negative/GND. In this system the power supply as up to 12V, so the 12V jumper was left and 5V was available on the module [23]–[27]. Also, Arduino GND is connected to pin 5 on the module as well to complete the circuit. Then six digital output pins on the Arduino were connected to the L298N Dual H Bridge DC Motor Driver,

two of which needed to be PWM (pulse-width modulation) pins. PWM pins were denoted by the tilde ("~") next to the pin number as shown in Fig. 7.



Fig. 7. Digital (PWM) pins on Arduino

Finally, the Arduino digital output pins were connected to the driver module. Digital pins D7, D8 and D9 were connected to pins IN1, IN2 respectively. Then, the EnA jumper is removed and D9 is connected to it. The Arduino digital input pins were connected to the Incremental Encoder, channel A with the digital pins D2 and channel B with digital pins D4. The wiring diagram of the Arduino and the L298N Dual H-Bridge DC Motor Driver and Incremental Encoder is shown in Fig. 8.

D. Software Design

The programming of the position control system was programmed in LabVIEW as depicted in Fig. 9. The final structure for the code is a loop. There are two functions outside the loop; the first one opens the VISA, where the Arduino input is specified by it. Also, the phase bode rate which is the speed of data transfer between the LabVIEW and the Arduino, it must be a large value to obtain high speed of data transfer. In general, speed of data transfer 9600 bit/sec and the number of byte that can be transferred in a second is 300 byte in a second which is too slow for the motor. When the speed one million bit per second is used, the number of byte that can be transferred in a second is 125000 byte in a second which is good for the motor work. The second function is closing function, which saves the running case after stopping the system and saving the last command as it before the closing.

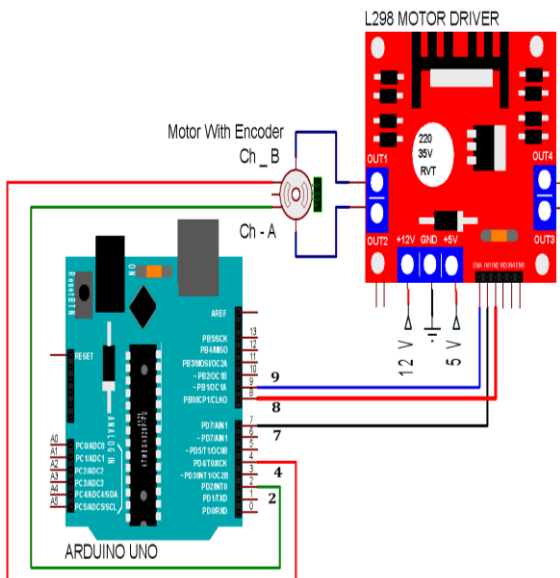


Fig. 8. System wiring diagram.

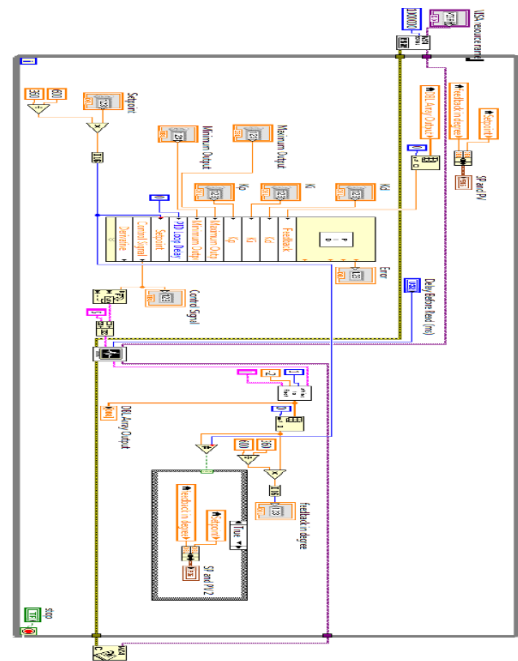


Fig. 9. Block diagram of the system with PID controller in LabVIEW

III. RESULTS AND DISCUSSIONS

This section discusses the results of PID that were applied to drive the DC motor to the desired position. The controller is needed to control the system by looking at the response of the system that specifies some criteria. To say the controller has did a good job for the system, one of the first things that must be done is to decide upon a criterion for measuring how good a response is. For example, it does not matter how the response reaches a steady state but the steady state itself either has a large error or not. However, transient behavior is also important in determining a good response. For that, some criterion will be introduced such as settling time, rise time, overshoot and steady state error. LabVIEW software was used to plot the output response of the system. Where, the desired position of the motor was set to be 120° .

This section demonstrates the results of a real time plotting of DC motor position control system using the PID controller with trial and error method. First of all, the values of the PID controller parameters was set to  $K_p=0.9$  ,  $K_i=0.058$  and  $K_d=16$ . Fig. 10 shows the output response of the position with these parameters.

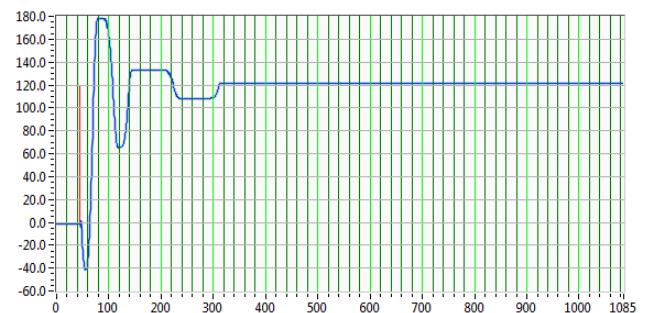


Fig. 10. Position response with  $K_p = 0.9$  ,  $K_i = 0.58$ , and  $K_d = 16$

It's clear that the system angular shaft position is  $120^\circ$ . The values of the controller parameters made the transient response has a high overshoot of about 19.44%, and the settling time is approximately 300 msec and the steady state error is zero. In order to improve the overshoot and the settling time, the controller parameters  $K_p$  and  $K_d$  have to be increased.

During trial and error, it was noticed that the most effective controller parameter on the system oscillation is the integral term. Therefore, the integral term is set to small value ( $K_i = 0.06$ ) to make the system more stable. Meanwhile, the derivative gain was increased to 60 in order to reduce the overshoot, as well as the settling time. The value of proportional gain was increased to be  $K_p = 6$  to make the system faster. Fig. 11 shows the real time plotting of DC motor position response when  $K_p = 6, K_i = 0.6, \text{ and } K_d = 60$ .

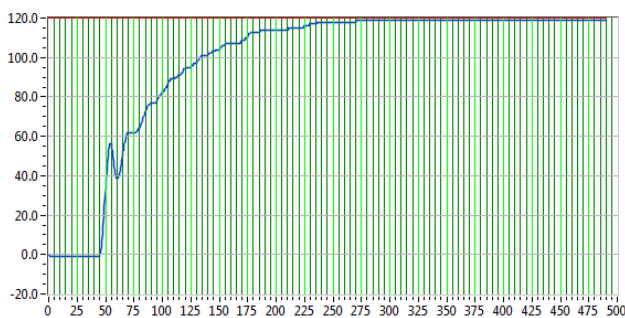


Fig. 11. Position response with  $K_p = 6, K_i = 0.6, \text{ and } K_d = 60$

The steady state error is still zero and the transient response has no overshoot, 100% with no oscillation. However, the output response has taken a long time to reach its final value with a high settling time approximately 260 msec. In order to improve the overshoot and the settling time, the controller parameters  $K_p$  has to be decreased and  $K_d$  has to be increased. By decreasing the values of the proportional term and increasing the derivative term, the new values of the controller parameters were  $K_p = 3, K_i = 0.058, \text{ and } K_d = 70$  and the position response is illustrated in Fig. 12.

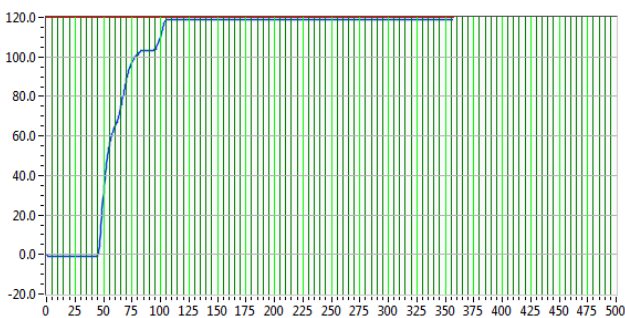


Fig. 12. Position response with  $K_p = 3, K_i = 0.058, \text{ and } K_d = 70$

With Reducing the PID controller gains to  $K_p = 3, K_i = 0.058$ , and increasing  $K_d$  to 70 caused the system to be relatively faster with a settling time of approximately 90 msec, the overshoot is zero and the steady state error is still zero too.

The designed PID controller is now tested in terms of setpoint tracking performance with same values of the controller parameters  $K_p = 3, K_i = 0.058, \text{ and } K_d = 70$ . The system is driven to track the setpoint changes in up and down stairs-like manner. By applying different setpoint changes at the input from  $120^\circ$  to  $160^\circ, 260^\circ, 360^\circ, 260^\circ, 120^\circ, 60^\circ$  and to  $0^\circ$ . The output response with setpoint tracking is shown in Fig. 13.

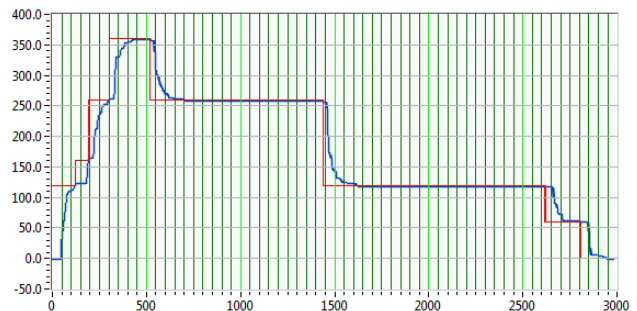


Fig. 13. Position response for set point tracking.

According to Fig. 13, it seems that the PID can track the commanded setpoint, with good performance in terms of both the transient and steady state response for the all change cases.

The noise signal is one of the unwanted input signals that can affect the output signal. An unwanted input signal often interferes with the control system [28]. The angular position of the DC motor can be affected by disturbances in the form of forces entering the control system. The feedback system can effectively reduce it, the block diagram of the system experiencing interference is shown in Fig. 14.

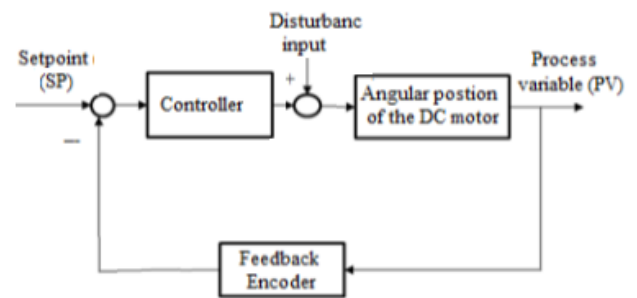


Fig. 14. Block diagram of position control system under effect of disturbance

To show the ability of the controller for the disturbance rejection that acting on the system. A force was applied on the angular position of the DC motors shaft to divert it from its desired position and see whether the controlled system can return the shaft to its desired position or not. When the position at the desired position ( $120^\circ$ ) on the angular shaft, there were two forces were applied on the position at different times as an external disturbance. The real time plot of angular shaft under the effect of these forces is shown in Fig. 15.

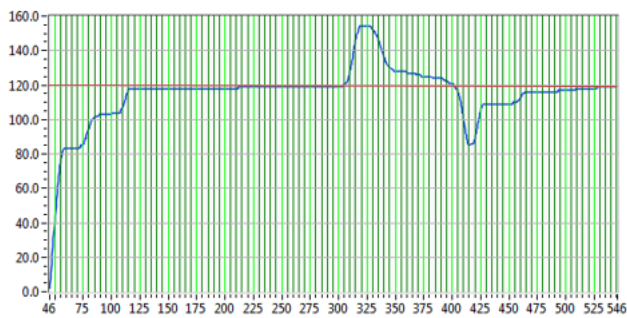


Fig. 15. Real time plot for the system under the effect of disturbance

External noise uses two forces that are applied to a position at different times. The angle axis under the influence of this force shown in Figure 15 is a plot of real time.

From the real time plot image, it can be seen that the system is affected by disturbances. There are two disruptions affecting the system at two different times. The first disturbance occurred at about 300 msec and the second disturbance effect at about 450 msec. The system can be stabilized again and can be returned to the angle axle to its original position in a few milliseconds with the PID controller. Interference can be rejected by using PID control.

#### IV. CONCLUSIONS

Three objectives in this study have been achieved, namely the real time model of the DC motor position control system has been successfully designed and developed, LabVIEW and Arduino are used to implement the PID control algorithm and succeed in the hardware system, and the motor output response is obtained using the labVIEW software. Control parameters can be adjusted from the experimental results. The LabVIEW based Arduino board was used to control the position of a DC motor whereby the simulation with LabVIEW software makes data analyzing more accurate, and saves research development time, as the motor position and generating the actuating signals to drive the DC motor and control its direction can be easily adjusted. L298N H-bridge driver was used to control the speed and the direction of the DC motor. PWM was programmed using Arduino to control the motor speed. The incremental encoders was used to provide the feedback signal. Where, the angular position of the encoder shaft was obtained by algebraic summing of the number of pulses provided by the encoder according to CW and CCW rotation. The direction of the rotation is determined by a digital decoding scheme using the two quadrature signals, and the direction changes are detected in an angular interval equal to a quarter of the angular step of the graduation. The transfer function of PID controller is used to implement the PID controller code in LabVIEW sub routine. Hence, the trial runs with different PID parameters were carried out several times in order to find the best system response. The best PID control parameters was chosen using trial and error tuning method, which are  $K_p = 3$ ,  $K_i = 0.058$ , and  $K_d = 70$  that gave the best response in terms of both transient and steady state response. Moreover, the system was tested for step tracking input where a different angle input signals were applied to the system in the running case and the results in the previous section showed that the designed

controller forced the output to track the desired input with short time and zero steady state error.

The performance of the controlled system are considered in presence of an external disturbance which is a force acting on the motor shaft and the results illustrated that the proposed PID controller performed satisfactorily in the rejection of disturbance. Hence, it can be concluded that the robustness of the proposed controller is investigated.

#### REFERENCES

- [1] Baćac N., Slukić V., Puškarić M., Štih B., Kamenar E., Zelenika S, "Comparison of different DC motor positioning control algorithms," 37th International Convention on Information and Communication Technology, Electronics and Microelectronics, pp. 1654–1659, 2014.
- [2] E. Kamenar and S. Zelenika, "Micropositioning mechatronics system based on FPGA architecture," 36th International Convention on Information and Communication Technology, Electronics and Microelectronics, pp. 138-143, 2013.
- [3] G. F. Franklin, J. D. Powell and A. Emami-Naeini, Feedback control of dynamic systems, 2nd ed, Addison-Wesley, 1991.
- [4] N. Kumar and P. Krishna, "Low cost data acquisition and control using Arduino prototyping platform and LabVIEW," IJSR, vol. 2, pp.366-369, February 2013.
- [5] A. I. Bin Man, Designed a real-time DC motor speed and position control for learning purpose using operational amplifier as an analog PID controller, Bsc. Project, UTMM, May 2008.
- [6] S. Raghavan, Digital Control for Speed and Position of A DC Motor, MS Thesis, Texas A&M University Kingsville, 2005.
- [7] I. I. Ince, A. Negrea, M. Imecs and C. Szabo, "Incremental Encoder Based Position and Speed Identification: Modelling and Simulation," Acta Universitatis Sapientiae Electrical and Mechanical Engineering, pp.27-39, 2010.
- [8] F. Arvani, S. N. Ferdous and M. T. Iqbal, "Digital control of MS-150. Modular Position Servo System," IEEE NECEC, November 2007.
- [9] J. Liu, P. Zhang and F. Wang "Real Time DC Servo Motor Position Control by PID Controller Using Labview," International Conference on Intelligent Human-Machine Systems and Cybernetics, pp. 206-209, August 2009.
- [10] R. Ranjani, R. Preethii, S. J. Sumitha, "Controlling the speed of a DC motor using LabVIEW," IJAREEIE, Vol. 4, pp. 9463-9469, December 2015.
- [11] N. H. Binti Yaziz, Digital speed and position control system incorporating an incremental encoder, M.Sc. project, UTHM, July 2014.
- [12] Ince, J. J., Szabó Cs., and Imecs M., "Modeling and simulation of an incremental encoder used in electrical drives," 10th International Symposium of Hungarian Researchers on Computational Intelligence and Informatics, Budapest, Hungary, pp. 97-109. 2009.
- [13] M. Ren, J. Zhang, H. Yue, and S. Zhou, "Adaptive neural network cascade control system with entropy-based design," IET Control Theory Appl., vol. 10, no. 10, pp. 1151–1160, Jun. 2016.
- [14] C. Jauregui, M. Duarte Mermoud, G. Lefranc, R. Orstica, J. C. Travieso Torres, and O. Beytia, "Conical Tank Level Control with Fractional PID," IEEE Lat. Am. Trans., vol. 14, no. 6, pp. 2598–2604, Jun. 2016.
- [15] A. S. Bazanella, L. F. A. Pereira, and A. Parraga, "A New Method for PID Tuning Including Plants Without Ultimate Frequency," IEEE Trans. Control Syst. Technol., vol. 25, no. 2, pp. 637–644, Mar. 2017.
- [16] J. Fei, Y. Chu, and S. Hou, "A Backstepping Neural Global Sliding Mode Control Using Fuzzy Approximator for Three-Phase Active Power Filter," IEEE Access, vol. 5, pp. 16021–16032, 2017.
- [17] J.-H. Yang and H.-K. Xu, "Robust Controller Design for Non-Minimum Phase UAV System and System Analysis," IEEE Access, vol. 6, pp. 70734–70769, 2018.
- [18] A. Bisoffi, M. Da Lio, A. R. Teel, and L. Zaccarian, "Global Asymptotic Stability of a PID Control System With Coulomb

- Friction,” *IEEE Trans. Automat. Contr.*, vol. 63, no. 8, pp. 2654–2661, Aug. 2018.
- [19] S. Han and S. P. Bhattacharyya, “PID Controller Synthesis Using a  $\sigma$ -Hurwitz Stability Criterion,” *IEEE Control Syst. Lett.*, vol. 2, no. 3, pp. 525–530, Jul. 2018.
- [20] J. Wang, Z. Luo, Y. Wang, B. Yang, and F. Assadian, “Coordination Control of Differential Drive Assist Steering and Vehicle Stability Control for Four-Wheel-Independent-Drive EV,” *IEEE Trans. Veh. Technol.*, vol. 67, no. 12, pp. 11453–11467, Dec. 2018.
- [21] T. MohammadRidha et al., “Model Free iPID Control for Glycemia Regulation of Type-1 Diabetes,” *IEEE Trans. Biomed. Eng.*, vol. 65, no. 1, pp. 199–206, Jan. 2018.
- [22] V. Gholamrezaie, M. G. Dozein, H. Monsef, and B. Wu, “An Optimal Frequency Control Method Through a Dynamic Load Frequency Control (LFC) Model Incorporating Wind Farm,” *IEEE Syst. J.*, vol. 12, no. 1, pp. 392–401, Mar. 2018.
- [23] J. Benavides, A. del Pozo, and E. Salinas, “Identification and control of a laboratory-scale prototype for crushing copper,” *IEEE Lat. Am. Trans.*, vol. 14, no. 2, pp. 549–558, Feb. 2016.
- [24] W. M. A. Rosado, L. G. V. Valdes, A. B. Ortega, J. R. Ascencio, and C. D. G. Beltran, “Passive Rehabilitation Exercises with an Ankle Rehabilitation Prototype Based in a Robot Parallel Structure,” *IEEE Lat. Am. Trans.*, vol. 15, no. 1, pp. 48–56, Jan. 2017.
- [25] N. Wang, X. Chen, Q. Lan, G. Song, H. R. Parsaei, and S.-C. Ho, “A Novel Wiki-Based Remote Laboratory Platform for Engineering Education,” *IEEE Trans. Learn. Technol.*, vol. 10, no. 3, pp. 331–341, Jul. 2017.
- [26] M. Rogerio Caldieri, J. Andre Bigheti, and E. Paciencia Godoy, “Implementation and Evaluation of Wireless Networked Control Systems using Modbus,” *IEEE Lat. Am. Trans.*, vol. 15, no. 2, pp. 206–212, Feb. 2017.
- [27] C. A. Calderon, C. Ramirez, V. Barros, and G. Punin, “Design and Deployment of Grasp Control System applied to robotic hand prosthesis,” *IEEE Lat. Am. Trans.*, vol. 15, no. 2, pp. 181–188, Feb. 2017.
- [28] Mustafa Saad, Mohammad Khalallah, “Design and implementation of an embedded ball-beam controller using PID algorithm,” *Universal Journal of Control and Automation*, vol. 5 pp. 63-70, 2017.