



Thu Dau Mot University  
Journal of Science

journal homepage: [ejs.tdmu.edu.vn](http://ejs.tdmu.edu.vn)



## DC Motor Speed Stabilizer System uses PID algorithm

by *Nguyễn Thành Đoàn, Phạm Hồng Thanh* (Thu Dau Mot University)

**Article Info:** Received 16 April, 2021, Accepted 2 June 2021, Available online 15 June 2021  
Corresponding author: [doannt@tdmu.edu.vn](mailto:doannt@tdmu.edu.vn)  
<https://doi.org/10.37550/tdmu.EJS/2021.02.198>

### ABSTRACT

*Automation or automatic control has been a growing trend in industrial fields as well as in life because of the advantages and utilities that help optimize work and reduce manpower in production. In medium and large scale automatic control systems, PLC (Programmable Logic Controller) is used as the control device for the system. In production, the transmission system using motors takes a large proportion and the problem of controlling for them to operate correctly, meeting technological requirements is one of the difficulties that need to be solved. In this study, we have applied PLC to stabilize motor speed according to preset value using PID algorithm integrated in Siemens S7-1200 PLC in order to apply research results in practice. At the limit of the article, we will present the issues of PID function block configuration, programming algorithm flowchart for PLCs, and monitoring console design for a small DC motor drive system model to verify.*

**Keywords:** *PLC, PID, motor speed stabilizer, DC motor, drive*

### 1. Introduction

Nowadays, computers and PLCs have become an indispensable part in the process of controlling machines in factories and factories. In machines or technological lines, a very important part is the electric motor (Deng et al., 2018; Hadi & Sallom, 2019). Besides, the problem of controlling its operating modes has many problems that need to be solved. The Proportional-Integral-Derivative (PID) is an algorithm studied and applied in many fields of stability control (Al Andzar & Puriyanto, 2019). In this

research, we will present about PID and apply this algorithm to control motor speed by Siemens S7-1200 PLC.

Besides the theoretical problems of PID algorithm (Elsrogy et al., 2013), PID programming application for S7-1200 PLC by TIA Portal software (Sartika et al., 2019), a hardware model to verify the results is also presented at the end of the study. The results of this study will contribute to learning about automation systems, PLCs, and practical simulation models for students and lecturers at Thu Dau Mot University.

## 2. PID algorithm

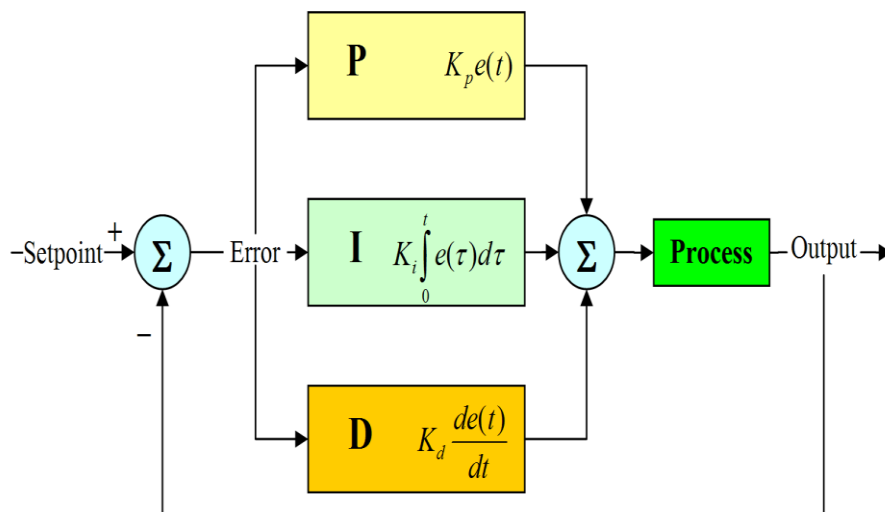
### 2.1. PID definition

A Proportional - Integral - Derivative (PID) controller is a controller using feedback loop control technique widely used in automatic control systems. The PID controller tries to correct the deviation between the input and output signals and then outputs a control signal to adjust the process accordingly. The PID controller is a controller consisting of 3 responses: proportional response, integral response and derivative response. It is a responsive closed control system that is used a lot in industry (Ang et al., 2005).

The proportional response ( $K_p$ ) has the effect of increasing the response rate of the system, and reducing, but not eliminating, the steady - state error of the system.

Integral response ( $K_i$ ) has the effect of eliminating established errors but can reduce the response rate of the system.

Derivative response ( $K_d$ ) increase system stability, reduce overshoot and improve system response speed.



**Figure 1.** PID controller block diagram

Considering the effects of the  $K_p$ ,  $K_i$ ,  $K_d$  on the closed system are summarized in the following table (Ang et al., 2005):

TABLE 1. Effects of increasing a parameter independently Table

Parameter	Rise time	Overshoot	Settling time	Steady-state error	Stability
$K_p$	Decrease	Increase	Small change	Decrease	Degrade
$K_i$	Decrease	Increase	Increase	Eliminate	Degrade
$K_d$	Minor change	Decrease	Decrease	No effect in theory	Improve if $K_d$ small

### 2.2 Proportional

The Proportional produces a control signal that is proportional to the value of the deviation. This is done by multiplying the error  $e$  by the constant  $K_p$  - called the proportional constant.

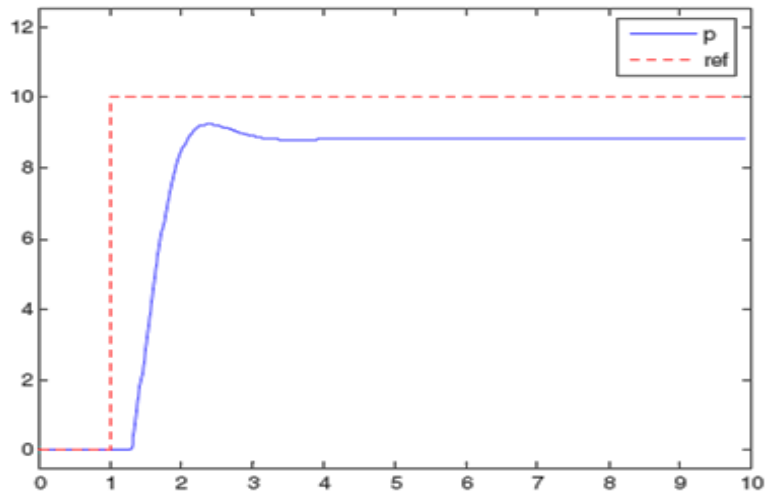


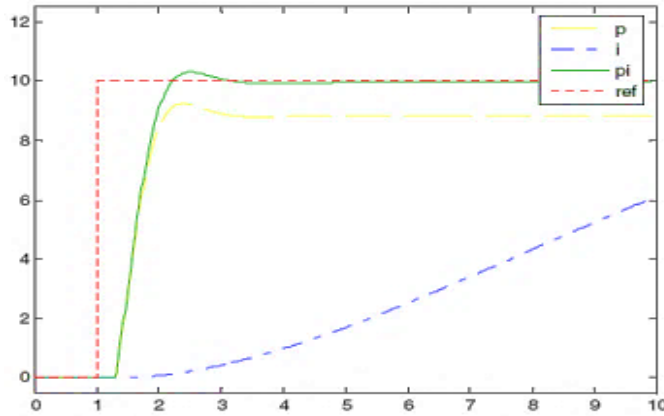
Figure 2. Response of Proportional

If only Proportional is present then in all cases static error always occurs, unless the input value of the system is 0 or already equal to the desired value. If the  $K_p$  value is too large, the system will be unstable.

### 2.3 Integral

Integral response adds the sum of the previous errors to the control values. The sum of the errors is carried out continuously until the obtained value is equal to the set value and as a result, when the system equilibrates, the error is zero.

Integral term usually comes with Proportional, which constitutes a PI controller. The Integral slows down the response of the system, while the PI eliminates the residual steady-state error.



**Figure 3.** Response of Proportional (P), Integral (I) and Proportional Integral (PI) controller

### 2.4. Derivative

I-controller doesn't have the capability to predict the future behavior of error. So it reacts normally once the set point is changed. D-controller overcomes this problem by anticipating the future behavior of the error. Its output depends on the rate of change of error with respect to time, multiplied by derivative constant. It gives the kick start for the output thereby increasing system response.

### 2.5. PID Controller

PID controller consists of three terms, namely proportional, integral, and derivative control. The combined operation of these three controllers gives a control strategy for process control. PID controller manipulates the process variables like pressure, speed, temperature, flow, etc. Some of the applications use PID controllers in cascade networks where two or more PID's are used to achieve control (Soe & San, 2019).

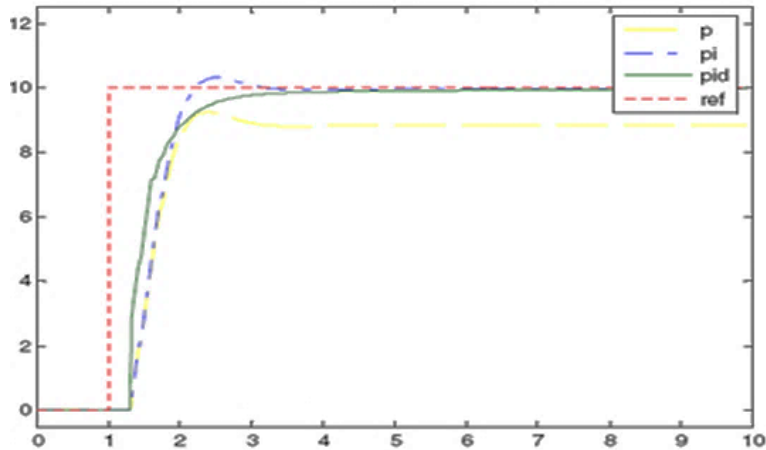
The figure 1 shows the structure of the PID controller. It consists of a PID block which gives its output to the process block. Process/plant consists of final control devices like actuators, control valves, and other control devices to control various processes of industry/plant.

A feedback signal from the process plant is compared with a set point or reference signal  $u(t)$  and the corresponding error signal  $e(t)$  is fed to the PID algorithm. According to the proportional, integral, and derivative control calculations in the algorithm, the controller produces a combined response or controlled output which is applied to plant control devices.

All control applications don't need all three control elements. Combinations like PI and PD controls are very often used in practical applications.

The differential equation of the ideal PID controller is below (Ang et al., 2005):

$$u(t) = K_p e(t) + K_i \int e(t) dt + K_d \frac{de(t)}{dt}$$



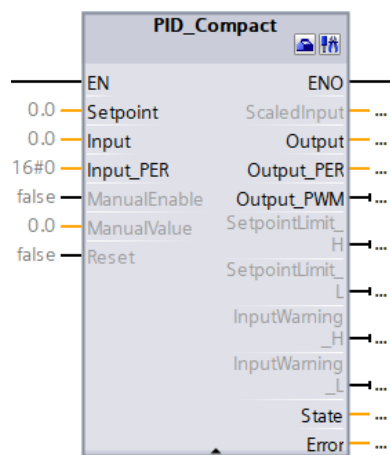
**Figure 4.** Response of PID controller (green curve)

### 3. PID\_compact function in TIA Portal


Uses: 1 PID\_Compact provide PID controller with self-tuning function for automatic mode or manual (Al Andzar & Puriyanto, 2019; Soe & San, 2019).

Note: It is necessary to create an OB30 cycle interrupt function block because the PID takes time to execute, it is not advisable to let the PID\_Compact block in the main program OB1 will cause the PLC's scan cycle to increase so much that the application has multiple sets The PID then the OB1 becomes slower and the response of the PLC is slow according to (Soe & San, 2019).

How to create a PID\_Compact function block: Choose Instructions → Technology → PID Control → Compact PID → Drag and drop PID\_Compact into the Network window.



**Figure 5.** PID\_Compact function in TIA Portal

To open the configuration window for the PID\_Compact function block, click the icon  on the PID\_Compact block.

In Basic parameters → Select Controller type as Speed → 1/min... → Input: select Input → Output: choose Output\_PER.

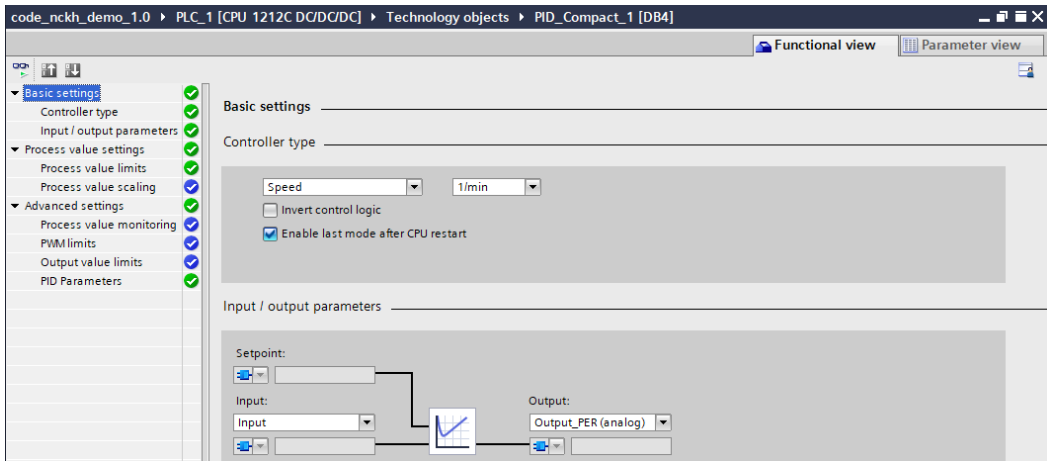



Figure 6. PID configuration window

In the Process value settings: enter parameters Process value high limit is the maximum speed of the motor, Process value low limit is the minimum speed of the motor.

In the Advance settings section, only pay attention to PID parameter → PID parameters: here the parameters can be adjusted manually or using automatic mode.

To enable PID in the Commissioning window, click on the icon .

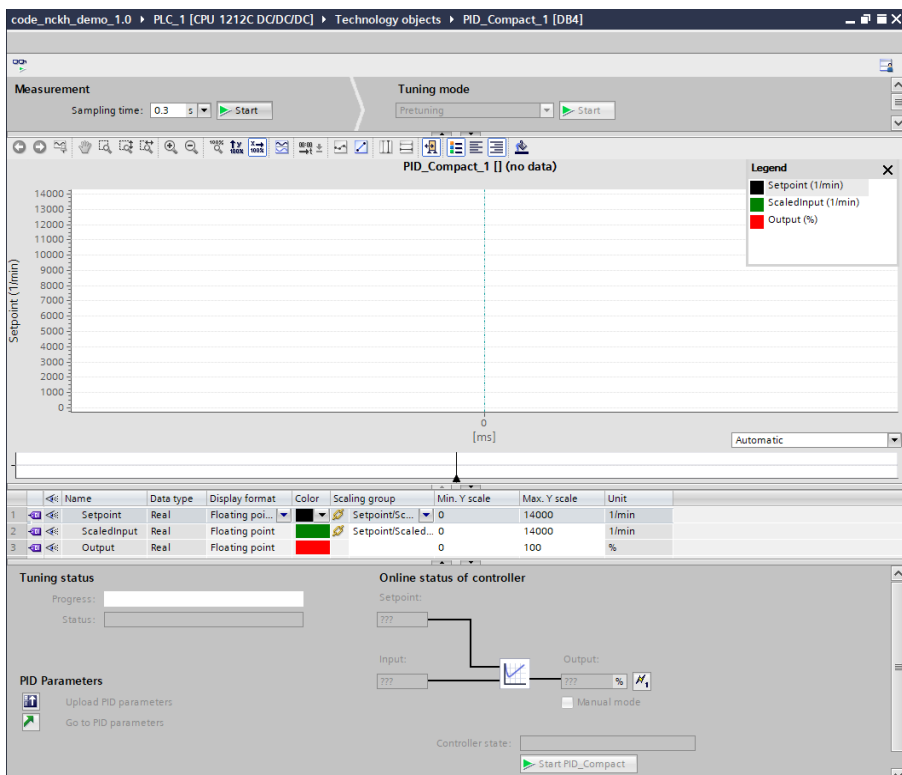


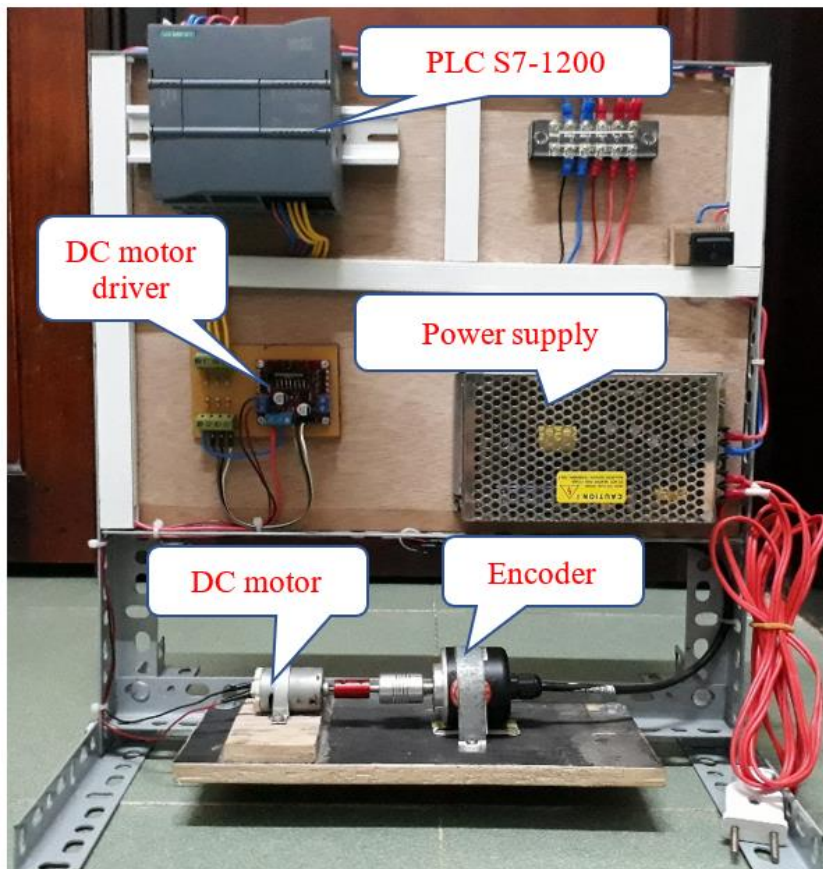
Figure 7. PID Commissioning window

Next step, we will start to select the sampling time, continue to select Fine turning to start to exploring parameters for the system. In the “Status” area the action steps and an error occurred will be displayed. Operation bar shows the progress of operation steps. If no error occurs, the PID parameters will be optimized for optimal adjustment. The PID controller switches to automatic mode and uses those optimal parameters. The optimum parameters are saved by pressing the (*Upload PID parameters*) button. And every time Power ON the PLC again, this parameter will be used.

#### 4. Practical PID Model and Result

##### 4.1 Practical Model for surveying

In this section, we will give a drive model with DC motor and related control parts to re-experiment the theory presented above in Figure 8.






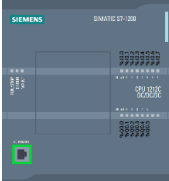
**Figure 8.** The practice model PID speed control for DC motor

Detailed descriptions of the components and parameters of the system are shown in Figure 9 below. The motor speed is read by the encoder and sent to the PLC through the High Speed Counter (HSC); meanwhile, the motor driven by the L298N Module

receives control signals from the PLC through the PWM pulse generator to determine the power to be supplied to the DC motor to achieve the setting speed.

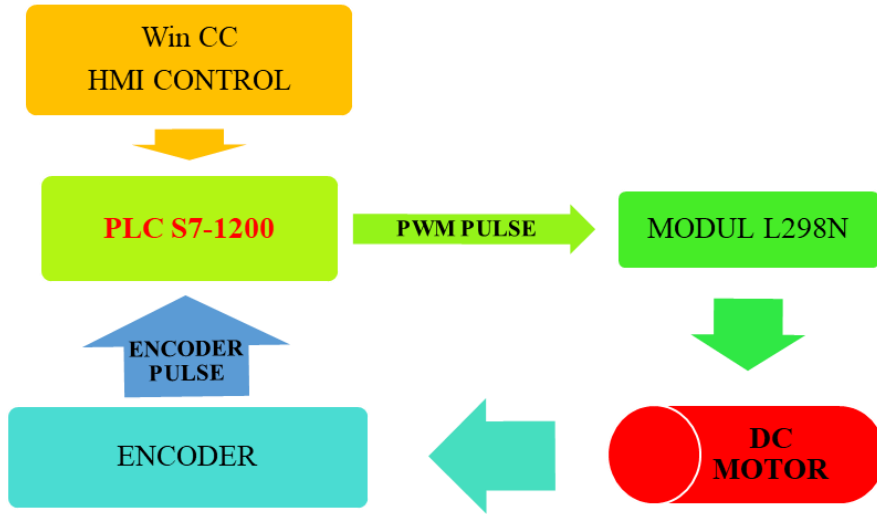
In this paper, the motor's speed parameters will be set through the Human Machine Interface (HMI) as well as monitoring the actual engine speed parameters during operation. Motor ON/OFF or Forward/Reverse control will also be performed on the HMI via PLC S7-1200.

TABLE 2. Technical Parameters of Main components

No.	Name & Image	Description
1	 <p><b>Incremental Rotary Encoder</b></p>	Technical parameters: + Model: E50S8-360-3-T-24. + Diameter of shaft: $\phi 8\text{mm}$ + Resolution: 360 pulses per revolution (ppr). + Output phase: A, B, Z. + Output: Totem pole. + Power supply: 12-24VDC $\pm 5\%$ .
2	 <p><b>Motor DC RS385</b></p>	Technical parameters: + Model: RS385. + Diameter of shaft: $\phi 2.3\text{mm}$ . + Length of shaft: 16.5mm. + Voltage 24VDC – Speed ~ 14000 rpm – No load current ~ 100mA.
3	 <p><b>Module L298N.</b></p>	Technical parameters: + IC: L298 - Dual Full Bridge Driver. + Voltage Input: 5~30VDC + Maximum Power: 25W for 1 Bridge. + Maximum Current Output for 1 H bridge: 2A. + Logic voltage: Low -0.3V~1.5V, High: 2.3V~Vss.
4	 <p><b>PLC S7 – 1200</b></p>	Technical parameters: + CPU: 1212C DC/DC/DC 6ES7 212-1AE31-0XB0. + I/O: 8 DI /6 DO (24VDC), 2 AI (0 – 10VDC) + Power supply: DC: 20.4V – 28.8V. + Memory: 50 KB. + High Speed Counter (HSC): 6. + Pulse generator (PTO/PWM): 4.

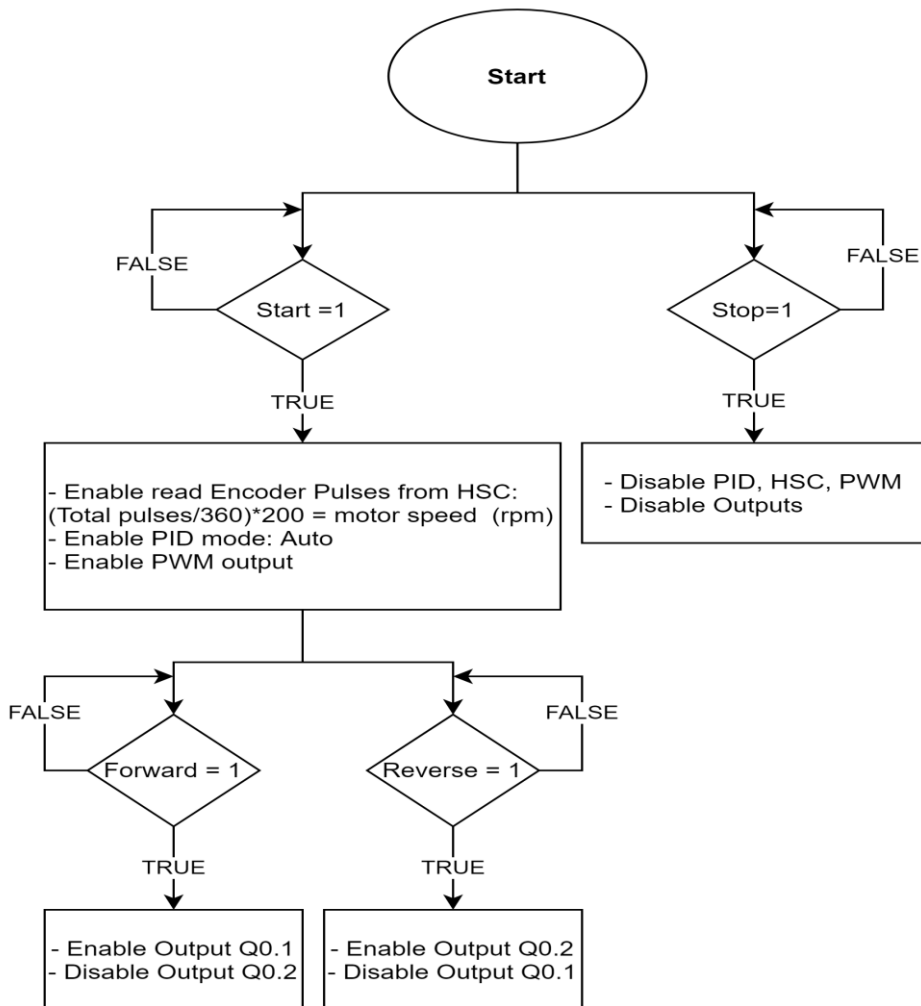
Block diagram describing the control and operating mode of the system is shown in Figure 10 below

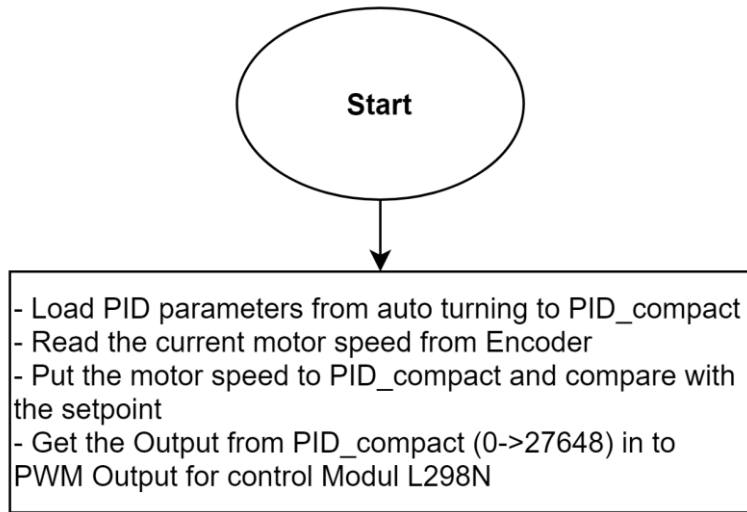




**Figure 9.** Block Diagram of the survey system

The process of the system is described in the following flowchart:

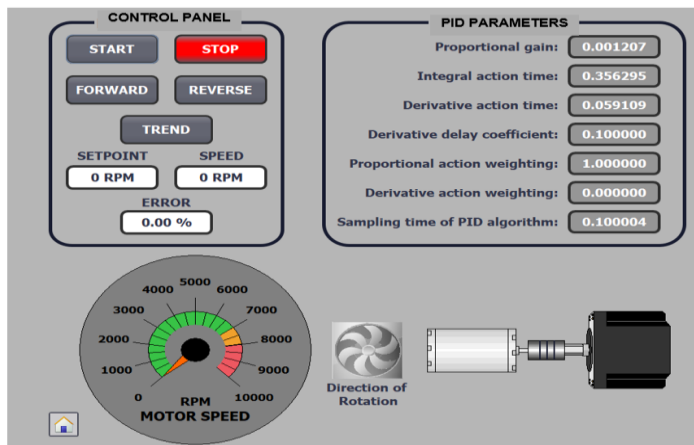




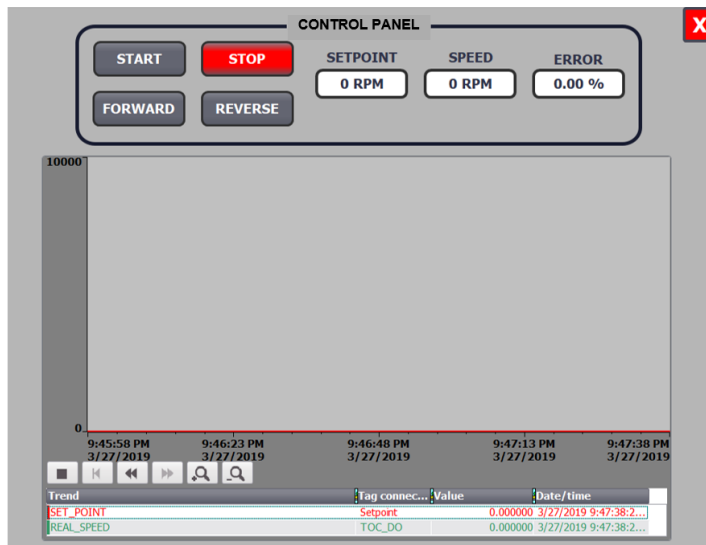
**Figure 10.** Block Diagram of Main and Interrupt Programe

*Main program:* When you press **START**, the system start to allow reading of Encoder pulses via high speed counter (HSC). Then, carry out a calculation to get the engine speed. Specifically, take the Encoder pulse in 0.3s divided by 360 (the resolution of the Encoder) then multiply by 200, so you get the motor speed in round per minute (rpm). At the same time when the **START** button is pressed, the PID controllers are allowed to operate in automatic mode and the PWM pulse generator is also allowed to pulse. When **FORWARD** is pressed, turn on L298N Module control output for forward rotation of motor. Similarly, pressing **REVERSE** button enables the motor to reverse.

*Interrupt program:* When the program starts, load the PID parameters detected in the PID\_Compact configuration. Get the actual speed calculated from the encoder pulse to the input of the PID set to compare the desired value entered to output the control signal. Take output\_per output control signal of PID unit (0 → 27648) and put in PWM to generate control pulse.



**Figure 11.** Main Interface



*Figure 12. Graphical Interface*

*Control and monitoring:* We design a SCADA interface that can be controlled and monitored on a PC or HMI monitor. Control part: turn on / off the system, select the motor direction of rotation and enter Setpoint (desired value); monitoring: display the detected PID parameters in the system, the actual motor speed, the error between the desired speed value and the actual speed, showing the motor's current direction of rotation (see Figure 10). In the graphical interface, in addition to the functions of turning on / off the system, selecting the direction of rotation and monitoring the real speed and error, there are more graphs to show the change of speed over time (see Figure 11).

#### **4.2 Result and Discussion**

*Results of the research:* In this paper, we have built a stable motor speed model using PID controller in PLC. The basic requirement for stabilizing engine speed is met according to the preset value. Easy-to-use control and monitoring interface, display current speed, compare actual value and desired value via graph. But the downside is that if the motor runs at a high speed, there will be noise, so the actual speed relative to the setpoint will fluctuate more than the low speed. The fix is to find other parameters for the controller to work better and use high resolution Encoder for more accuracy in speed measurement.

*Future Research:* We will design controller for AC motor drives with larger capacity or variable torque in conveyor applications, plastic injection machines,... Besides, the PID parameters set in this paper is fixed during operation, every change of system parameters needs to be adjusted, which is time-consuming and unsuitable for cases where system parameters change. Therefore, our next research direction will develop a PID control system with adaptive changing parameters to meet the system requirements

(Elsrogy et al., 2013; Pérez et al., 2014).

## References

- Al Andzar, M. F., & Puriyanto, R. D. (2019). PID Control for Temperature and Motor Speed Based on PLC. *Signal and Image Processing Letters*, 1(1), 7–13. <https://doi.org/10.31763/simple.v1i1.150>
- Ang, K. H., Chong, G., & Li, Y. (2005). PID control system analysis, design, and technology. *IEEE Transactions on Control Systems Technology*, 13(4), 559–576. <https://doi.org/10.1109/TCST.2005.847331>
- Deng, M., Mao, J., & Gan, X. (2018). Development of Automated Warehouse Management System. *MATEC Web of Conferences*, 232, 1–6. <https://doi.org/10.1051/mateconf/201823203051>
- Elsrogy, W. M., Fkirin, M. A., & Hassan, M. A. M. (2013). Speed control of DC motor using PID controller based on artificial intelligence techniques. *2013 International Conference on Control, Decision and Information Technologies, CoDIT 2013*, 196–201. <https://doi.org/10.1109/CoDIT.2013.6689543>
- Hadi, H. H., & Sallom, M. Y. (2019). Pneumatic Control System of Automatic Production Line Using SCADA Implement PLC. *4th Scientific International Conference Najaf, SICN 2019*, 15(3), 37–42. <https://doi.org/10.1109/SICN47020.2019.9019356>
- Pérez, I. G., Godoy, A. J. C., & Godoy, M. C. (2014). Fuzzy controller based on PLC S7-1200: Application to a servomotor. *ICINCO 2014 - Proceedings of the 11th International Conference on Informatics in Control, Automation and Robotics*, 1, 156–163. <https://doi.org/10.5220/0005013601560163>
- Sartika, E. M., Sarjono, T. R., & Saputra, D. D. (2019). Prediction of PID control model on PLC. *Telkomnika (Telecommunication Computing Electronics and Control)*, 17(1), 529–536. <https://doi.org/10.12928/TELKOMNIKA.v17i1.11589>
- Soe, Y. Y., & San, P. E. (2019). *Pid closed-loop control analysis for automation with siemens plc using tia v13*. 2(August), 200–208.