Proportional integral derivative DC motor speed control system for xy plane movements

IOP Conf. Series: Materials Science and Engineering **508** (2019) 012082 doi:10.1088/1757-899X/508/1/012082

Proportional integral derivative DC motor speed control system for xy plane movements

Hadian Satria Utama*, William Wijaya and Joni Fat

Electrical Engineering Department, Faculty of Engineeringb Universitaas Tarumanagara, Jln. Letjen. S.Parman No.1, Jakarta 1140, Indonesia

*hadianu@ft.untar.ac.id

Abstract. DC motors are the most widely used electrical motors since their ease of use. Controlling a DC motor can be done by adjusting amount of voltage and current supplied to the motor. In this design the controlling is done using proportional integral derivative (PID) control, applied for movement in XY plane. Ziegler-Nichols method is used to determine PID constants. A Raspberry Pi is used to receive and process input from the user through a graphical user interface (GUI). The data input consists of two values, i.e. position and velocity. The GUI also shows the actual coordinate position on the screen. Two motors are controlled simultaneously by the Raspberry Pi through serial peripheral interface (SPI) with one microcontroller for each motor axis. The program contains a PID control algorithm to stabilize the motor speed. The coordination of the two stabilized motor speed is used to draw a sloping line. Sloping lines are drawn in order to determine the performance of the PID controller.

1. Introduction

Electrical motors are the most widely used electromechanical device in industry. It works by converting electrical energy to mechanical energy. The resulting mechanical energy can be used for rotating screws, pump impellers, fansor blowers, driving compressors, lifting materials, etc. There are two types of motors based on the electrical current used to drive the motors, i.e. alternating current (AC) motors and direct current (DC) motors. DC motors are more widely used since their rotation speed is easier to control compared to AC motors [1].

DC motor speed is controlled by varying the applied electrical voltage or current. Motor controlling is used so that the motor is able to be moved as desired, which in turn will be used for moving objects with a certain level of precision. Besides controlling motors to move in a particular speed, it must also have all basic controls in controlling motors such as start, stop, and completed with functions able to protect motors, machines, products and operators [2].

A lot of research has been done to create a motor controller system. The most frequently applied are fuzzy logic and proportional integral derivative (PID) method. Between the two methods, PID control is more widely used in the industry [3]. Based on all things above, then a motor controller using PID control method would be applied in the movements on xy-plane coordinates.

Comparison to other research was done before designing this system [4], [5]. The designed system is a DC motorcontrol with PID. The motor is then connected to the two motors to be used for moving the point coordinates in xy-plane. The moving point coordinates will leave a line like pattern. The line was measured. Its dimension was measured and compared to calculation result of two point coordinates entered. The ratio determines whether the PID controller works well or not. Other than that, the LCD display will show moving point coordinates when running. The point movement was also limited by point movements limit, limited by movement limiter module which would stop the motor when the point exceeds the maximum limit of allowed movements.

2. Design

The DC motor speed control system designed controls the coordinate point movements on xy-plane of the designed machine. The moving point coordinates draws a line with dimensions matched the input data entered by users. The line drawing with a desired gradient is performed by controlling the speed ratio of x and y-motor speed. The motor speed ratio can be expressed in the gradient equation of a straight line, i.e.:

$$\frac{Vy}{Vx} = m. \quad (1)$$

$$\frac{Vy}{Vx} = \frac{y_2 - y_1}{x_2 - x_1} \quad (2)$$

Dengan, v_y = motor speed in y-axis direction (m/s)
v_x = motor speed in y-axis direction (m/s)
X1 = value of x-axis coordinate of line starting point (m)
x_2 = value of x-axis coordinate of line ending point (m)
y_1 = value of y-axis coordinate of line starting point (m)
y_2 = value of y-axis coordinate of line ending point (m)

Equation (2) is applied to the drawing program by calculating the gradient of the origin point and the destination point. From equation (1), gradient and speed entered by the user are used to calculate the speed in each axis. The speed value entered by the user will be the highest speed value. A gradient value of larger than one (m>1) will cause y-axis motor speed becomes speed value entered by the user, while a gradient value of smaller than one (m<1) will cause x-axis motor speed becomes speed value entered by the user. The only exception occurs is when the straight line has zero or infinity gradient.

The motor speed is controlled using PID algorithm, which is an algorithm for reaching set point (SP) speed by calculating the difference between the actual value and the desired value. This value is calculated and stored in current error and previous error variables before it is multiplied by the constants. The PID constants are K_P, K_I dan K_D. Although there are only three constants but in order to determine them, systematic steps are needed, since anerrorwill cause the system to become unstable. The PID system isclosed loop [7].The block diagram of the closed loop system is shown in Figure 1.

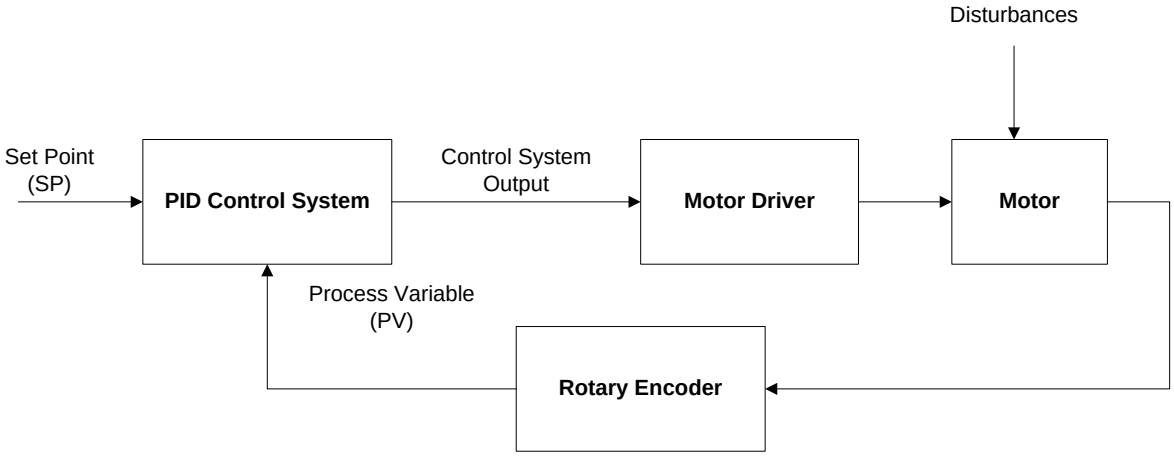


Figure 1. PID Block Diagram

The speed entered from the controller would be the SP, the purpose of the control system is for maintaining the motor speed at SP with the help of feedback from the rotary encoder. The value read from the encoder is called process variable (PV). The controlling will be done by calculating the error value from the difference between SP and PV. This error values are then added to produce the response. The system response is then sent to the motor driver to drive the motor.The PID algorithm is entered into the microcontroller. The PID algorithm needs to be changed first from continuous to discreet, so that it can be implemented in the program. Generally, the PID method equation is shown as follows:

$$u(t) = K\left(e(t) + \frac{1}{T_i}\int_0^t e(t)dt + T_d\,\frac{de(t)}{dt}\right) \quad \text{.....(3)}$$

with, $u(t)$ = control system outputas a function oftime
$e(t)$ = difference between SP and PV
K = constant
T_i = integral period
T_d = differential period

Equation (3) is the PID equation for continuous system. In order to use it in discreet system, it should be transformed using Laplace and z-transform that can be derived as follows:

$$u(t) = K_p e(t) + K_i\int_0^t e(t)dt + K_d\frac{d}{dt}e(t) \quad \text{.....(4)}$$

with $K_p = K$, $K_i = \frac{K}{T_i}$, $K_d = KT_d$

Equation (4) is then Laplace transformed to:

$$G(s) = K_p + \frac{K_i}{s} + sK_d \quad \text{.....(5)}$$

The result of Laplace transform is then z-transformed to be used in discreet system. The result of the z-transform is as follows:

$$U(z) = \left[K_p + \frac{K_i}{1-z^{-1}} + K_d(1-z^{-1})\right]E(z) \quad \text{.....(6)}$$

Rearrangement of equation (6) gives:

$$U(z) = \left[\frac{(K_p+K_i+K_d)+(-K_p-2K_d)z^{-1}+K_d z^{-2}}{1-z^{-1}}\right] E(z) \quad (7)$$

By introducing new variables that contains each constant value, the following are obtained:

$$K_1 = K_p + K_i + K_d \, , K_2 = -K_p - 2K_d \, , K_3 = K_d$$

Equation (7) can be rewritten to:

$$U(z) - z^{-1}U(z) = [K_1 + K_2 z^{-1} + K_3 z^{-2}]\, E(z) \quad (8)$$

By performing inverse z-transform, the following equation is obtained:

$$u(t) = u[t-1] + K_1 e[t] + K_2 e[t-1] + K_3 e[t-2] \quad (9)$$

Equation (9) is ready to be applied to discreet system which will be programmed with

$u(t)$ = current control system output
$u[t-1]$ = control system output before $u(t)$
$e[t]$ = current error
$e[t-1]$ = error occurs before $e[t]$
$e[t-2]$ = error occurs before $e[t-1]$
K_1 = $K_p + K_i + K_d$
K_2 = $-K_p - 2K_d$
K_3 = K_d

The program segment for the microcontroller written based on equation (9) is as follows:

```
1  void PID()
2  {
2   if (input_kecepatan!=0) setpoint = input_kecepatan;
4  e2=e1;
5  e1=e;
6  pv=abs(rpm);
7  e=setpoint-pv;
8  outputpwm= outputpwm + k1*e + k2*e1 + k3*e2;
9  if(outputpwm>OUTPUTMAX)
10  outputpwm=OUTPUTMAX;
11  else if(outputpwm<OUTPUTMIN)
12  outputpwm= OUTPUTMIN;
13  OCR1A=(int)outputpwm;
14  }
```

The PID procedure is called each time the motor rpm value is updated. *The* rpm value in line 6 is obtained from counting external encoder interrupt called by Overflow Interrupt Timer 0 every 120 ms. The program changes the PWM output values by changing the OCR1A value. In line 8, change in PWM depends on the values of k1, k2, and k3 obtained from the tuning result using Ziegler-Nichols method added to the previous PWM output. PWM mode used in the program is the Phase Correct Mode. Line 9-12 limits the PWM output value so that it does not exceed the maximum value that can be stored in OCR1A register. The flowchart is shown in Figure 2.

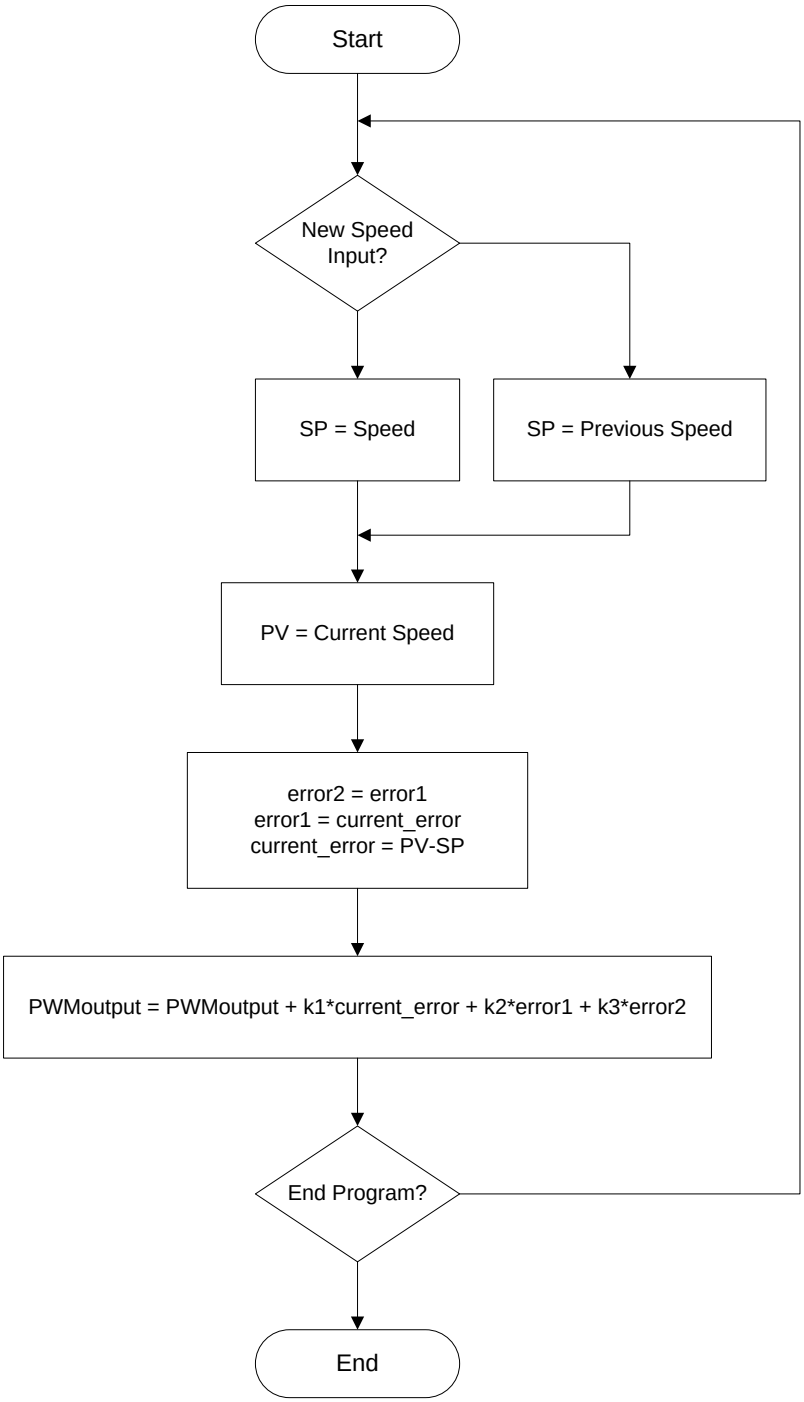


Figure 2. PID software flowchart

The PID constants selection used is the Ziegler-Nichols method. This method has been applied in a lot of industrial machinery PID control and a lot of other PID methods are developed based on this method. The Ziegler-Nichols method gets the tuning values from experiments simulated in a real system or simulated system [8].

The tuning procedure consists of five steps, i.e.:

1. The value in SP is assigned with a value that is the machine's working value. Then, make the process in the system approaching SP by controlling the PV value until the difference is 0.

2. Change the PID controller to P controller by assigning values to T_I=0 and T_D=0.
3. The K_P value is increased slowly until the system oscillates. The K_P instantaneous value before oscillation is measured. The measured value is called critical gain (K_C).
4. The next value measured isthe oscillation periode (T_C). This value is obtained by recording the time when oscillation first occurs.
5. After the two values are obtained, the PID parameter values are determined based on the Ziegler-Nichols table shown in Table1.

Table 1. Ziegler-Nichols formula for closed loop method controller [8]

Controller Type	**K_P**	**T_I**	**T_D**
P *Controller*	0.5 K_C	0	0
PI *Controller*	0.45 K_C	0.833 T_C	0
PID *Controller*	0.6 K_C	0.5 T_C	0.125T_C

The values $K_I = 1/T_I$ and $K_D = 1/T_D$ are obtained from Table 1.

The drawn line data is entered by the user using the graphical user interfaces (GUI) made in the Raspberry Pi. The display program displays the machine coordinate point and the actual machine condition. The inputs of the program are sent through SPI communication to operate the machine. The program is written in Python 2.7.1 language. The widgets in the GUI are made using GTK+3 modules with the help of Glade Interface Designer3.14.2 software. The GUI display is shown in Figure 3.

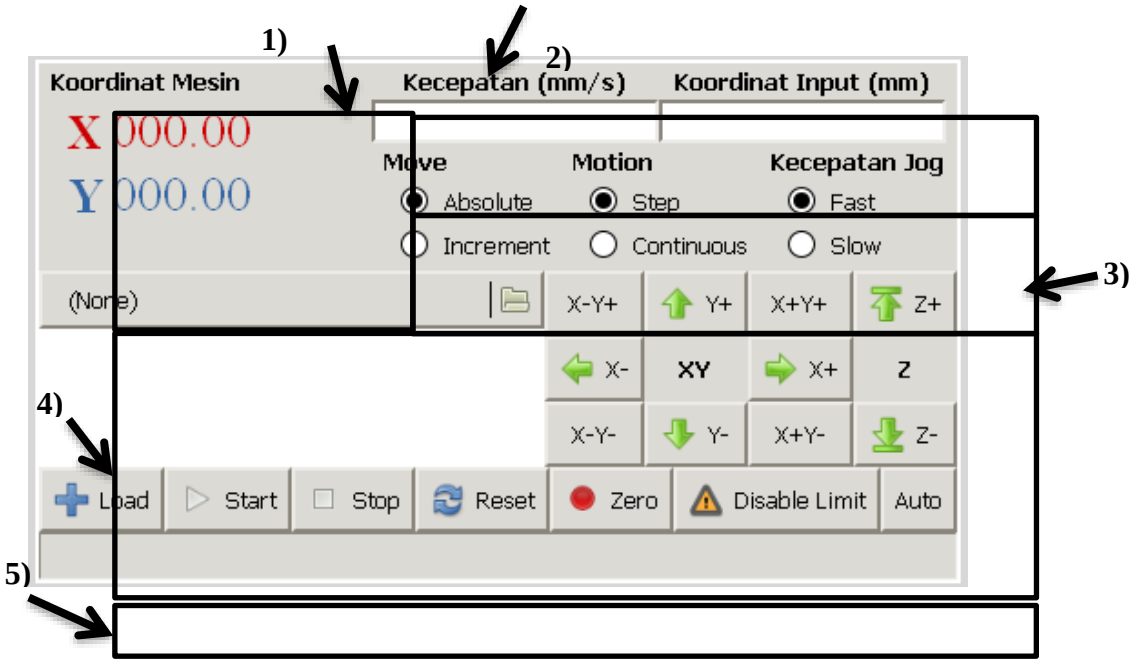


Figure 3. GUI of the machine

The GUI of the machine consists of several rooms, i.e.:

1. Display room
2. Input room

3. Option Button room
4. Button room
5. Status Bar room

The display room updates the coordinate values of the machine every 100 ms. The input room accepts inputs from user and checks the correctness of input format. The speed format must be a positive number less than 100 mm/s, while the coordinate input format is “x value, y value”. The values of x and y entered must be less than 1000.

The radio button room is provided for selection of modes to be used which matches the frame title. The radio button move group is used for controlling the coordinate movement type. Selecting absolute mode will send the destination value of the machine according to the coordinates entered. The increment mode is used for sending the destination value of the sum ofcurrent position value and input coordinate value. The selection of radio button motion group is used for controlling the pushed jog button. Selecting step mode will run the machine when one of the jog or direction button is pushed and stop the machine when it is released. Continuous mode selection runs the machine when one of the jog button is pressed and would only stop when the stop button is pressed. The radio button group jog speed controls the machine movement speed when it is moving using the jog button.

The button room is used for controlling the machine, both automatically and manually. The program runs in two modes, i.e. auto and manual. When auto button is selected, all buttons that is running the machine directly and related radio button will not function, while when manual button is selected, start button, stop button, movement control radio button speed value input, and coordinate will not function. Status bar room displays machine condition. The function of each part is shown in Figure 4.

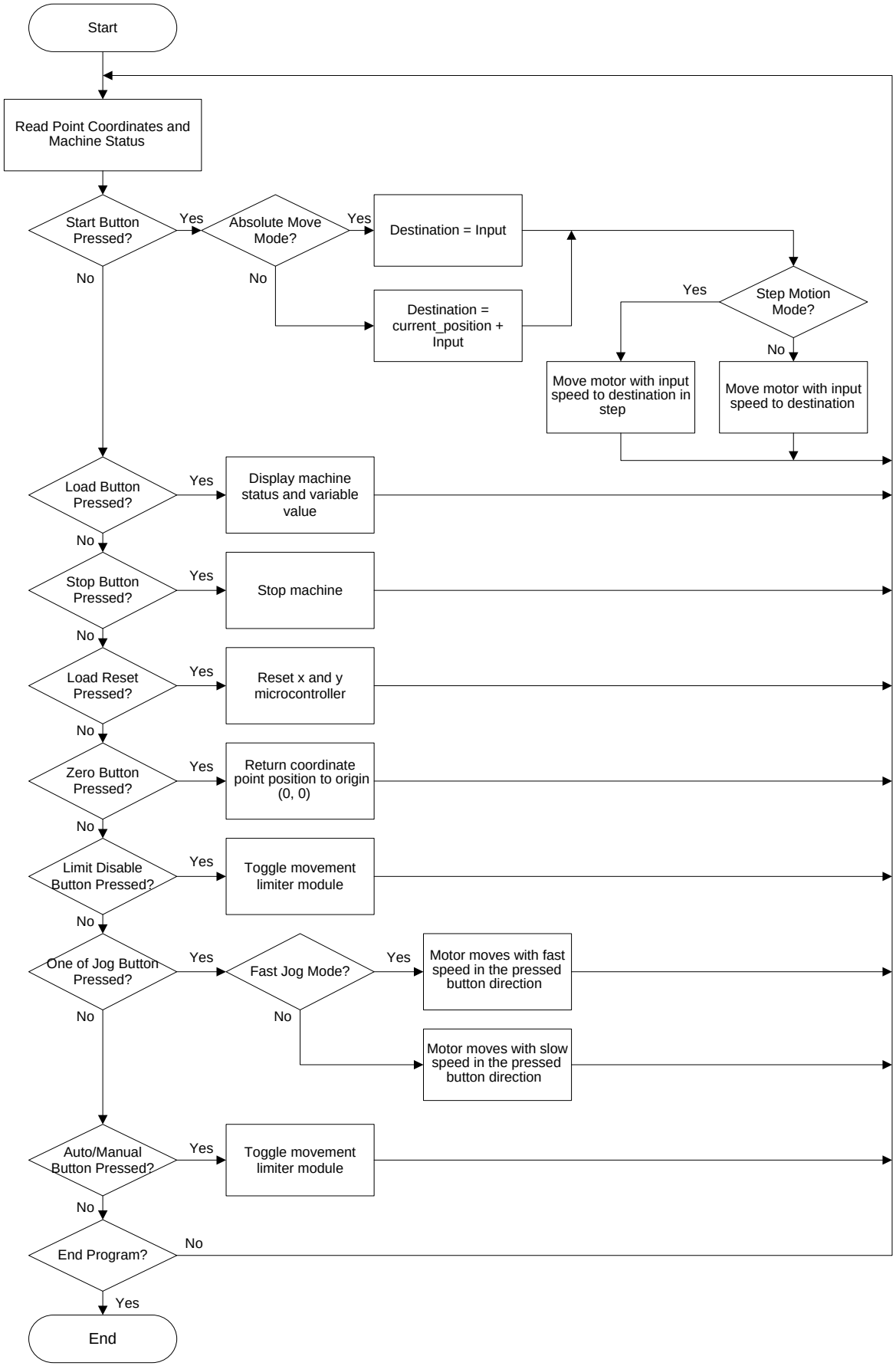
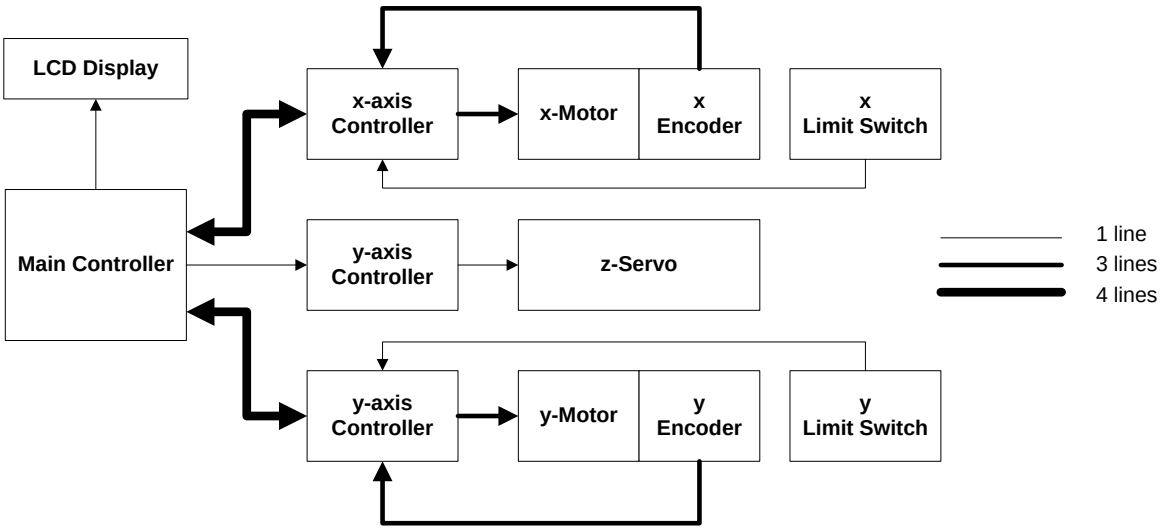


Figure 4. Program flowchart for display and input

Based on the concept description above, a block diagram is provided in order to show the interconnection between modules and make the design easier. The block diagram is shown in Figure5.



Gambar 5. System block diagram

3. Testing and Analysis

Testing was performed on each module to simplify troubleshooting when a failure in a particular module causes system malfunction. The overall system was performed after all the modules passed the test and the modules are connected together. Therotary encoder module testing is performed by rotating the motor manually in particular positions and counting the number of pulses produced. The motor was rotated to 45^0, 90^0, 180^0, and 360^0angles.A simple program in the microcontroller is used for counting the number of pulses from the encoder. The rotary encoder used in the design produces 500 pulses per cycle.The rotary encoder readings was recorded in Table 2.

Table2. Rotary Encoder Test Results

Angle of Rotation (^{0})	Number of pulses read from rotary encoder
45	64
90	124
180	251
360	500

Based on Table 2, the pulse reading is then converted back to degrees using the following equation:

$$Position = \frac{pulses\ read}{number\ of\ pulses\ in\ one\ cycle}\ x\ 360^0 \quad \text{.....(10)}$$

Using equation (10), the position obtained from the first reading was 46.08^0. The second reading was 89.28^0, the second reading was180.72^0, and the last reading was 360^0. The maximum percentage average of reading error is 2.4% on first reading. The errors occurred because of parallax error caused by the view angle that was not orthogonal.

The DC motor testing was performed to make sure it worked properly. The testing was performed by measuring maximum rotation speed and current used by the DC motor. Maximum rotation speed was obtained by connecting the motor directly to the 24 V_{DC} power supply. The test results were the DC motor maximum rotation speed used by the x-motor was 892 RPM and for y-motor is 776 RPM. The total current consumption for the 2 motors was 3.5 A. That amount of total current is within the capacity of the power supply module.

Limit switch was tested by measuring its resistance. Limit switchused is the normally closed type. Limit switch is good if the resistance is close to 0 (zero) when it is not pushed. On the other hand when it is not pushed, the resistance value is very large. The test results gave a resistance value of 15.72 kOhm when not pushed, and 0 Ohm when pushed.

The motor driver module testing is performed in two parts. In the first part the PWM output was given by the microcontroller to the motor and the RPM result was measured. The rotation direction and stopping of the motor was also performed in the first part. The test is successful if the motor speed increases when the given PWM value increases. The result of motor driver module testing controlling the DC motor speed is shown in Table 3. The motor driver must also be able to change the rotation direction of the motor and stop the motor. The motor driver module test result in driving the DC motor is shown in Table 4.

Tabel 3. Motor driver module on speed control test result

DC Motor	Motor driver PW input	Rotary encoder reading (RPM)
Motor X	50	114
	100	187
	250	325
	500	566
	1000	887
Motor Y	50	89
	100	109
	250	244
	500	476
	1000	758

Tabel 4. Motor driver module on movement control test result

Driver Motor Input		Motor movements
***Input* 1**	***Input* 2**	
0	0	Stopping without braking
0	1	Rotating in clockwise direction
1	0	Rotating in counter clockwise direction
1	1	Stopping by braking electronically

The second stage of testing is performed by using the PID algorithm in the microcontroller for controlling motor speed in a certain set point. The testing would be successful when the PID algorithm was able to maintain the motor speed and performance criteria was sufficiently good. The testing was performed by using PID constants obtained by trials. The PID response graphic is shown in Figure 6.

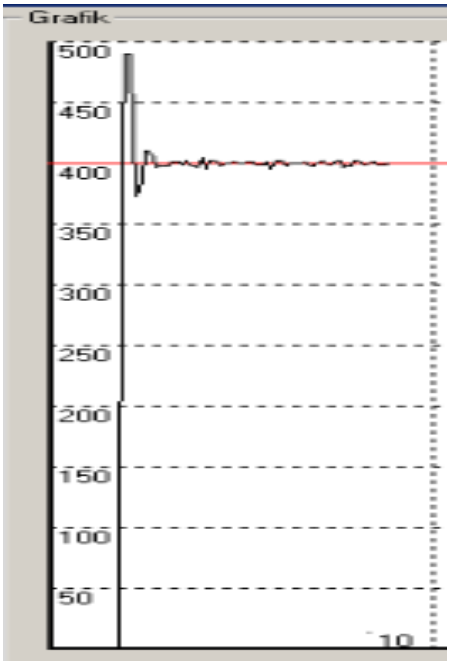


Figure 6. PID responsegraphic

The graphic was drawn by sending the processed data to a computer through serial communication. The computer program used was written using Visual Basic 6.0software. The PID constants used are K_P=1, K_I=0.24, and K_D=0.11. These values were obtained using the Ziegler Nichols method in Table 2, Kc = 1.67 and Tc = 0.24. The motor driver with PID modul test result is shown in Table 5.

Table 5. Motordriver with PIDmodul test result

Set Point	Speed when stable(rpm)	Rise Time(seconds)	Settling Time(seconds)	Steady State Error(%)	Overshoot(%)
100	105	0,3	1	5	28
200	196	0,5	1,4	2	24
300	298	0,8	1,5	0,6	23
400	401	1	1,9	0.25	20

As seen in Table 5, the overshoot values do not show good performance criteria, i.e. 5%. This will affect significantly if the movements performed is movements per millimeter, but the overshoot value does not affect movements in cm. All others achieved good performance criteria, i.e. maximum rise time of 1 second, maximum settling time of 2 seconds, and maximum steady state error of 5%.

Overall system testing was performed to make sure that the whole system work properly when all modules are integrated together. It started by entering the desired point coordinates input through the GUI. The input data was then sent to each controller andthe z-plane servo motor moved the plate down. The x and y axis motor controller moved the DC motor at the desired speed. The movements drew a line on the work plate. The motor stopped when the desired point coordinates was reached and the z-servo motor lifted the work plate up. The overall system test result is shown in Table 6. As seen in Table 6, there is some error between the drawn line and the input data. The maximum error is 1.3 mm as displayed on theLCD display, using a ruler it is1 mm. The sloping line with gradient had jagged edges since the mechanical part for Y axis experiences vibration when the motor is moving. The accuracy level obtained in this design is ±1 mm. The test results shows that the designed system is able to work as expected, the motor can be controlled to drive movements in both axis and has the correct dimension just like the data input to the system.

Table 6. Complete system test result

Input (mm)		Speed Input (mm/s)	Drawn Line	Display (mm)		Measuring Result (mm)	
X	Y			X	Y	X	Y
0	100	40	Parallel toy-axis	0	100,78	0	100,5
100	100	40	Parallel to x-axis	100,34	0	100	0
100	100	10	Sloping line with gradient 1	99,44	101,03	99,5	101
-100	-200	10	Sloping line with gradient2	-99,3	-199,8	-99	-99,200

4. Conclusion

Based on the test performed on the designed PID based motor speed control system, it can be concluded that among others:

1. The motor speed can be controlled until the speed is stable using PID. The motor stable condition can be reached in less than 2 seconds on average and the error when the condition is stable is less than 5% of the required speed. The performance criteria that is not achieved yet in the PID control is the maximum overshoot of the motor speed control system, as it exceeds 5%.
2. The accuracy of the designed system is ± 1 mm, since the microcontroller controls the speed every 120 ms. Therefore, if the speed used is 10 mm/s, then the maximum error occurred when the motor rotates before it is stopped is 1.2 mm.
3. The drawn sloping line is not straight and has jagged edge, since y-axis part vibrates when it moves. The pen connected to z-axis would be affected by the vibration. Therefore, sometimes it does not touch the working table. The vibration also cause the drawn line to be not straight since the y-motor speed becomes difficult to control.

5. References

[1] H Wayne Beaty and J Kirtley, *Electric Motor Handbook.* New York: McGraw-Hill, 1998, p. 3.
[2] S L Herman, *Electric Motor Control,* 9th ed. Delmar: Cengage Learning, 2010, p. 15.
[3] A Bagis, *Determination of the PID Controller Parameters by Modified Genetic Algorithm for Improved Performance: Journal of Information Science and Electronic Engineering 23, 1469-1480*. Turkey: Erciyes University, 2007, p. 1469.
[4] N Rachmadyanti, A Wijayanto, E Satriyanto, R Rokhana, *Kontrol PID untuk Pengaturan Kecepatan Motor pada Prototype Ayunan Bayi Otomatis.* Politeknik Elektronika Negeri Surabaya, Surabaya, 2012, hal 1-5.
[5] H Wicaksono, J Pramudijanto. *Kontrol PID Untuk Pengaturan Kecepatan Motor DC Dengan Metode Tuning Direct Synthesis Jurnal Teknik Elektro Vol. 4, No. 1, Maret 2004.* Institut Teknologi Sepuluh Nopember. Surabaya, Surabaya, 2005, hal 10-16.
[6] Ogata, Katsuhiko. *Teknik KontrolAutomatik I*. Jakarta: Penerbit Erlangga. 1994, p. 20.
[7] R S Burn. *Advanced Control Engineering.* Oxford: Butterworth-Heinemann, 2001, p. 5.
[8] J G Ziegler and N B Nichols. *Optimum Settings for Automatic Controllers.* Rochester: A S M E, 1942.