International Journal of Innovative Research in Engineering & Management (IJIREM)
ISSN: 2350-0557, Volume-8, Issue-6, November 2021
https://doi.org/10.55524/ijirem.2021.8.6.18
www.ijirem.org

Novel Design of Power Position Control Feedback System in MATLAB

Zahoor Ahmad Parray
M.Tech Student,
Department of Electrical Engineering, RIMT University, Mandi Gobindgarh, Punjab, India

Dr. Satish Saini
HOD, Department of Electrical Engineering, RIMT University, Mandi Gobindgarh, Punjab, India

Dharmendra Kumar
Assistant Professors,
Department of Electrical Engineering, RIMT University, Mandi Gobindgarh, Punjab, India

ABSTRACT
Motor position control is employed in a variety of settings, including the workplace, the home, and manufacturing plants. The motor load is expected to account for about 70% of total electrical consumption. The motors utilised in this section have been used for a wide range of tasks, including traction loads, rotating machines, and position control. Printing heads in printers, hoist motors, 3D printers, CNC machines, and even more modern robotic surgical equipment use position-controlled motors to communicate with the environment are all examples of position control. Position control is simply defined as a motor with a position encoder, such as a resistive encoder or an optical encoder that gives control signals to the controller, and a controller that regulates the power supplied to the motor using a power controller, such as a DC-DC regulator. In this project position control of a DC motor is done by using the Proportional-Integral-Derivative (PID) approach in Simulink. PID attempts to fix the disparity by calculating the error between the actual and desired locations and then creating a PWM (pulse width modulated) signal by which position can be adjusted. The system changes into a closed-loop when a PID algorithm is applied to it. Error developed by the DC motor as well as the motor's orientation can be corrected to the desired position when PID controller is combined with DC motor. The PID values Kp, Ki, and Kd are fine-tuned using Z. Nichols principles and the t&e approach, and to manage the direction of the DC motor using optimum values the PID algorithm is fine-tuned.

Keywords
Controller(PID), Closed Loop, Open Loop

1. INTRODUCTION
Electrical energy is transferred to mechanical energy with the help of a motor. Either DC sources or AC sources can be used to power an electric motor, because it had been widely used in industry sectors, robot manipulators, and electrical devices that require speed and position control, the DC motor was chosen for this dissertation. Developing dc motor applications is simple and adaptable since dc motors come in a number of shapes and sizes. It is very affordable and has good reliability and validity. [1]Controlling the position of a motor is crucial in a variety of situations. Such as robotics, where it is applied to control the placements of end effectors and arms, industrial control valves, and flood gate management at hydroelectric dams. They can also be utilized in-camera gimbals and other similar devices. Many other types of conventional control schemes have been developed to reduce load effects, including proportional-integral, proportional-integral-derivative (PID), adaptive, and robust controllers. Although the fact is that each of these methodologies have advantages and disadvantages in terms of utilization, the majority of controllers must be designed utilizing the plant's properties and detailed structure. As load effects arise, good control performance can't be reached if this is not done. As a result of this research, a control framework has been developed to mitigate the effects of large and/or unbalanced loads. [2]Because there are so many switching control methods, pulse width modulation (PWM) is being used as the control approach (J. A .Ramirez, Jan. 2001). Because of its simplicity, this control is widely employed on converters.

2. LITERATURE REVIEW
For efficient ultra-high precision control concepts and implementation, the accuracy of system models is important. As a result, the dynamics of each element must be properly understood. This can be aided by system identification techniques. The model's configuration should be as brief as possible, and ideally linked to the machine's physical characteristics (e.g., drive shaft compliance or carriage inertia). Several computer programs provide algorithms for assessing several model orderings and finding which one is the best fit for a given data set (e.g., MATLAB). However, this numerical method should not be used instead of a detailed understanding of the process or physical system. [3]

Two duties will be done by a microcontroller in a Feedback control system:

To evolve the desired velocity, on a regular basis transpose the average powers given to the engine. To determine the exact location and height of the output motor shaft. A sequence of pulses will be provided to the microcontroller's internal counter by the shaft encoder to represent rotational movement. Two tasks will be done by a timer configured for this cause in every 1/10th of a second (which is just an arbitrary value). One of these two software operations is to transpose the load angle of the shaft or the frequency. [4]

3. OBJECTIVES
My project's objectives are as follows:

- Assemble a motor position control system.
- Change the position of an actuator motor that uses the same setup.
- To improve the effectiveness of a PID controller in the same setup. This entails utilizing established algorithms to tune the PID controller.
- To maximize the setup's technical specifications, such as overreach, rise time, and settling time.

4. SYSTEM ANALYSIS
Designing a motor model using typical electrical engineering equations is the layout provided in this project. The design procedure then moves on to getting the position output from the motor model by differentiating the speed output of the motor

model. Combining this with set-point yields the desired output position profile. The PID controller smooths out the output and eliminates faults such as excessive overshoots and delayed responsiveness, among other factors. The motor model and PID controller are examined first, followed by a demonstration and description of the Simulink model for this project. After presenting and analyzing the facts, a conclusion is reached after modelling the arrangement. [5]

4.1. Creating a Motor Model

In control systems, DC motor is a normal actuator. When used in collaboration with wheels, drums, and wires it yields immediate rotating motion and additionally brings translational motion.

4.2. Physical Model of the Motor

As indicated in fig.1 we considered the armature circuit element and rotor force diagram (FBD) of a DC motor. V is the input voltage of armature in volts (run by a DC or AC voltage source). The angular velocity of motor shaft is calculated in rad / s, while the shaft's elevation is considered in θ.

Where:

Va = voltage of armature (V) Ra = resistance armature (Ω)

La = inductance of armature (H) ia = current of armature (A)

Eb= back emf (V) T = Torque (Nm)

θ = elevation of rotor shaft (rad)

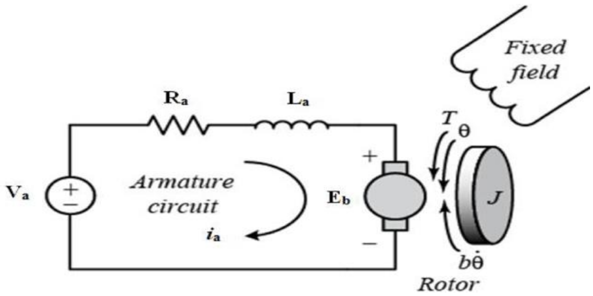


Figure 1: A diagram of the DC motor under consideration

Without affecting the voltage applied to the field, The appertained voltage to the stator of an unconventionally energized DC motor is reshaped in armature control, where the voltage at the output end and motor circulatory force(T) are connected by the expression below:

$$V_a(t) = R_a\, i_a(t) + L_a \frac{d\, i_a(t)}{dt} + Eb(t)$$

The motor circulatory force, T, is proportionate to the current of armature, allied by a constant K

$$T = Ki$$

The angular velocity is allied to the counter electromotive force (emf), Eb by:

$$E_b = K\,\omega = K \frac{d\,\theta}{dt}$$

As adorned in fig.3.1, the ensuing equations modeled on Newton's law and Kirchhoff's law may be drafted below:

$$J \frac{d^2\theta}{dt^2} + b \frac{d\theta}{dt} = Ki$$

$$L \frac{di}{dt} + Ri = V - K \frac{d\theta}{dt}$$

However, with the intention of using Simulink, the archetype must be a transfer function, which is acquired by putting in the Laplace transform to the previous two equations. As an outcome, we get the succeeding equations:

$$s(Js + b)\theta(s) = K\,I(s) \qquad \text{(a)}$$

$$(Ls + R)I(s) = V(s) - Ks\theta(s)$$

Character s typify the Laplace operator. I(s) can be scripted as follows:

$$I(s) = \frac{V(s) - Ks\theta(s)}{R + Ls}$$

To get the answer, plug everything into eqn (a).

$$s(Js + b)\theta(s) = K\, \frac{V(s) - Ks\theta(s)}{R + Ls}$$

Fig. 2 depicts the expression for a dc motor using a block diagram. We acquire the basic nonfeedback system function by removing I(s) from the preceding two equations, where the input is the voltage of armature and the output is the rotational speed.

$$P(s) = \frac{\theta'(s)}{V(s)} = \frac{K}{(Js + b)(Ls + R) + K^2}$$

Let's presume a motor in steady-state which functions at 0.1 rad/sec for an input of 1 volt. After all, running at the right speed is the fundamental necessitate of a motor, the deviation of speed of motor from the desired speed of the motor will be kept less than 1%. The motor should acquire the full speed (steady-state speed) as it is turned ON, this point is also considered in motor modeling. 2 seconds settling time is an obsession in this process. We just require about 5% overrun step response in this case, because system may fail at a higher speed than desired speed.

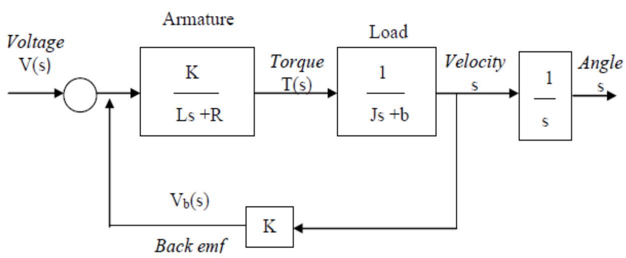


Fig 2: Expression of DC motor

In a nutshell, for a unit step command in motor speed the control system's output must follow succeeding points.

• It should take less than two seconds for the system to settle.
• The overrun rate should be less than 5%.
• Error rate of less than 1% under stable conditions

This is a list of all the components that will be used in the project's design. [6]

4.3. PID Controller

The P element adds a readjustment to get you near to the specified point. The more you get away from it, the more correction it applies. It only works in the present tense. The I term provides some memory to the adjustment. I apply more adjustment the longer you've been away from the set point. Over time, it cleans up mistakes. The swan, which is slow and elegant, is the I word. The knee jerk response is the D word. Any system shock results in a large opposing shock correction from D. It's only for a short time. In boot camp, the D word is the jittery sergeant major. It's difficult to fine-tune them. If and when that becomes of interest, the Ziegler technique provides a means to automatically calibrate the terms.

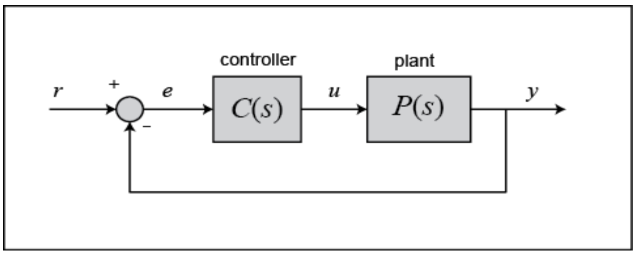


Fig 3: PID Controller

ISSN: 2350-0557, Volume-8, Issue-6, November 2021
https://doi.org/10.55524/ijirem.2021.8.6.18
www.ijirem.org

PID is in the form of a closed loop system, the time domain output of a PID controller is determined by feedback error, output of a PID controller in time domain is identical to the control Input of the system.

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

Let's start by looking at how the PID controller functions in a closed-loop system, as shown in the picture above. The difference between the anticipated output (r) and the actual output (e) is the tracking error (e) (y). The PID controller receives the error signal (e) and calculates the error signal's gradient and integral over time (t). The proportional gain (Kp) multiplied by the magnitude of the error, the integral gain (Ki) multiplied by the integral of the error, and the derivative gain (Kd) multiplied by the derivative of the error give the control signal (u) to the system. The controller collects this control signal (u), and the new output (y) is acquired. The new output (y) would be sent back into the system to be compared to the same reference signal and a second error signal generated (e). Based on the latest error signal, the controller computes an update to the control input. This operation will keep going as long as the controller is switched on.

The Laplace transform of Equation before is used to derive the system function of a PID controller.

$$K_p + \frac{K_i}{s} + K_d s = \frac{K_d s^2 + K_p s + K_i}{s}$$

Here
Kp = proportionality component, Ki = integral component, and Kd = derivative component. [7]

4.4. PID Terms

When the proportional component (Kp) is increased for almost the same degree of imperfection, the control signal is proportionately amplified. Since the controller will "push" more than that for a given level of error, the closed-loop system will respond directly, and will also overshoot more. A further impact of raising Kp is that steady-state error is significantly reduced, although not fully eliminated.it becomes easy to detect the mistake by improving derivative component (Kd). Just one way the control will improve is if the error rises, If Kp remains static under classic proportional control. Derivative control can cause the control signal to expand in size, even if the error is still small but the error tends to tilt upwards. By dampening the system this preemptive action reduces overshoot. Steady state error doesn't vary by the addition of a derivative component (Kd). [8]

Table 1: Closed loop effect on the overshoot, rise time, settling time, steady-state error

CL RESPONSE	RISE TIME	OVERSHOOT	SETTLING TIME	S-S ERROR
Proportional gain(Kp)	Diminish	Rise	Slight Change	Diminish
Integral gain(Ki)	Diminish	Rise	Rise	Diminish
Derivative gain (Kd)	Slight Change	Decrease	Diminish	No effect

4.5. PID Tuning Techniques

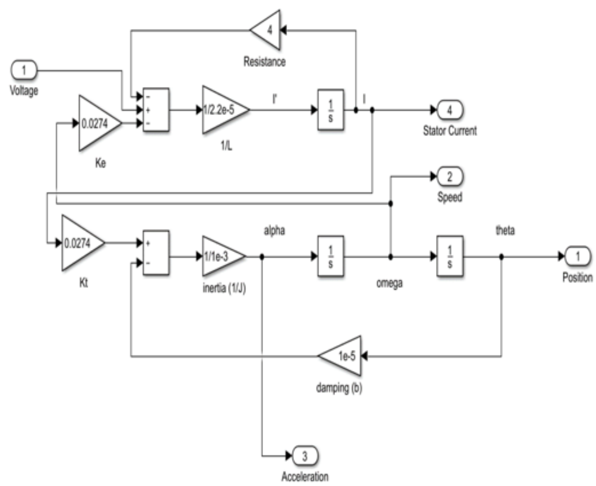
In order to build a PID controller for our system, the following points must b considered to obtain the desired response:

- We must acquire an open loop response in order to detect what requirements are to be done.
- Enhance the rising time by adding a proportional control.
- To minimize the overshoot a derivative control is to be added
- To lessen steady-state error, choose an embedded control system.
- In order to get desired response you must adjust Kp, Ki, and Kd, until you reach your goal. From the table on this "PID Tutorial" page you should choose a controller which will meet your requirements. You can collaborate all the three controllers into a single controller because all these are not required. There is no need to build a separate derivative controller (as in the above example) if all characteristics are met by a single PID controller. Little complexity in the setup must be maintained. An evidence of changing a PID controller on a real-world system may be found at the adherent link. When it comes to integrating control, control concentration, integrator wind-up, and noise intensification are just a few of the challenges that this situation poses.

MATLAB includes the ability to automatically determine appropriate PID gains, avoiding the requirement for the above-mentioned trial-and-error procedure. You can use "pidTuner" to access the tuning algorithm directly, or you can use "pidTuner" to access the tuning algorithm via a lovely graphical user interface (GUI). PID tuning technique using MATLAB can be used to choose PID components to maintain the steadiness of the conduct of the system (response time, bandwidth) and resilience (stability margins). 60 degree phase margin is automatically produced by this technique. This technique is also used to alter the PID controller in this project, and it works in the opposite direction. [9]

5. SIMULATION DESIGN

The motor model is the first step in the design process. The same is seen in the diagram below.
As illustrated below, this model is then implemented in a closed loop configuration:



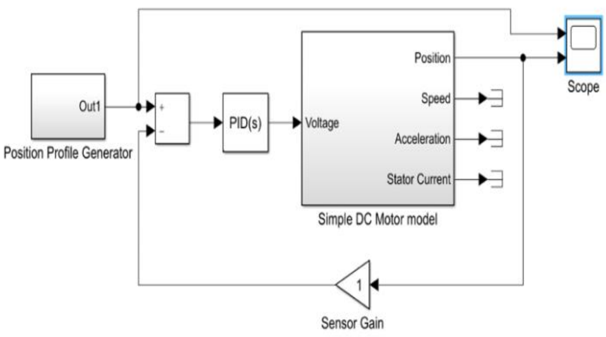


Figure 4: Closed Loop Configuration

The position sensor is simulated using the feedback gain.
The position profile generator simulates a real-world scenario in which a motor must transition between several locations in a short period of time, such as a CNC or 3D printer. [10]
The following are the motor constants:
J = 3.228E-4. b = 3.507E-4
K = 0.029. R = 4
L = 2.75E-4

6. RESULTS AND OBSERVATIONS

Simulink was used to simulate the model. As a result, the following result is obtained:

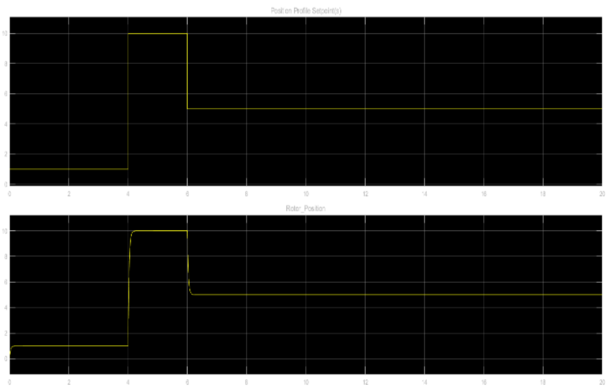


Figure 5: Position Profile Reference and Actual Motor Response

As can be observed, the motor strongly resembles the set-point profile within the previously mentioned performance constraints.
The upper plot is the position profile reference, and the lower plot is the actual motor response.

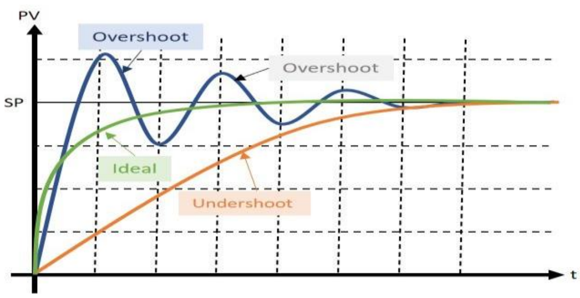


Figure 6: Behaviour of PID Controller under several condition of PID parameters

In this work we observed that by varying PID parameters we can easily adjust the position of motor shaft to our desired position (set point), we studied operation of motor by considering different values of PID parameters separately and found that PID controller can be much more efficient for controlling the position of DC motor than a PD or PI controller. Hence PID controller improves the efficiency of the system, but overshoot is noticed when lowering steady-state error to zero. Enhance the derivative gain to decrease overshoot, but the rising time will also rise as a result. As a result, there is a trade-off between overshoot and response time, which signifies we must sacrifice one to improve the other.

7. CONCLUSION

As a result, the model seems to work as expected. Trying to scale up the system and including control for AC and BLDC motors, as well as observer control techniques, could yield more results. Inverters and associated protection techniques can be added to the system as well. The system's stability can then be tested in a variety of scenarios, including malfunctions.

REFERENCES

[1] Chandra Shekhar Gohiya, S.S. Sadistap, S.A. Akbar, B.A. Botre Central," Design and Development of Digital PID Controller for DC Motor Drive System Using Embedded Platform for Mobile Robot", 978-1-4673-4529-3/12/31.00 c 2012 IEEE, 2013 3rd IEEE International Advance Computing Conference (IACC).

[2] "Control System Engineering" by I.J. Nagrath and M. Gopal, New age International, 2010.

[3] A.Chowdhury , D.Debnath ,NIT Agartala ," DC Motor Position Control using State Space Technique " , ISSN 2348-5426 , International Journal of Advances in Science and Technology (IJAST).

[4] Jose Carlos Gamozo-Real , Ernesto Vazquez-Sanchez and Jaime Gomez-Gil , " Position and Speed Control of Brushless DC Motors Using Senseless Techniques and Application Trends ", ISSN1424-8220.

[5] Mohamed Farid Bin Mohamed Faruq, University Malaysia Pahang, "PID Controller Design for Controlling DC Motor Speed Using MATLAB Application".

[6] Andrew Klee B.S., University of Central Florida, 2003, "development of a motor speed control system using MATLAB and Simulink, implemented with a digital signal processor".

[7] Benjamin J. Engle , Advisor: Dr. John Watkins ,Wichita State University , " MATLAB Based DC Motor Control Lab"

[8] N. Bacac, V. Slukic, M. Puskaric, B. Stih, E. Kamenar, S.Zelenika ,University of Rijeka, "Comparison of different DC Motor Positioning Control Algorithms".

[9] J. Villagra, B. Vinagre, I. Tejado, Data-driven fractional PID control: application to DC motors in flexible joints, Proceedings of the IFAC Conference on Advances in PID Control PID'12, Brescia(Italy), 2012

[10] C.I. Muresan, S. Folea, G. Mois, E.H.Dulf, Development and implementation of an FPGA based fractional order controller for a DC motor, Mechatronics, 23(7), pp. 798-804, 2013.