DIRECT SYNTHESIS BASED TUNING OF PID CONTROLLER DESIGN FOR SPEED CONTROL DC MOTOR

[1] Arun Yadav, [2] Maneesh Kumar Gupta, [3] Dr. Rajnish Bhasker
Department of Electrical Engineering, U.N.S.I.E.T.V.B.S.P.U Jaunpur U.P.
[1]arun02yadav10@gmail.com, [2]manishcripton@gmail.com, [3] rbhasker@vbspu.ac.in
https://orcid.org/0009-0002-0319-0535

***Abstract* - A PID controller in series with a lead/lag compensator is created using the direct synthesis approach for controlling open loop integrating processes. Consideration is given to set-point weighting in order to lessen the undesired overshoot. In the direct synthesis approach, guidelines are given for choosing the desired closed loop tuning parameter and set point weighting parameter. Significant load disturbance rejection performances are provided by the approach. To demonstrate the effectiveness of the proposed approach, DC motor speed regulation is taken into consideration. When compared to recently published approaches, a significant improvement is obtained. MATLAB/Simulink is used for designing simulation of proposed system.**

***Keywords*— PID, Direct Synthesis, DC motor, Simulink, Closed loop system**

I. INTRODUCTION

PID (Proportional-Integral-Derivative) control is a widely used technique for regulating and maintaining desired performance in various systems. When applied to DC motors, PID control plays a critical role in achieving precise speed or position control. This comprehensive guide explores the principles of PID control for DC motors, its implementation techniques, tuning methods, and optimization strategies[1,2]. By understanding the intricacies of PID control and its application in DC motor systems, engineers and enthusiasts can enhance motor performance, stability, and responsiveness in a wide range of industrial, robotic, and automation applications[3].

PID speed control of a DC motor refers to the use of a PID scheme to regulate and maintain the desired rotational speed or velocity of the motor [4]. It is a closed-loop control technique that continuously monitors the motor's actual speed, compares it to the desired setpoint speed, and adjusts the motor's input voltage or current to minimize the error and maintain accurate speed control.

The PID controller uses three components: the proportional, integral, and derivative actions. Each component contributes to the overall control signal sent to the motor to correct any deviation from the desired speed [5]. The proportional Integral Derivative controller's objective is to minimalize the fault among the desired and actual speeds by dynamically adjusting the switch sign.

The proportional component of the PID controller produces a control signal proportional to the difference between the desired speed (setpoint) and the actual speed (feedback) [6-7]. It acts as a corrective force that scales the control signal based on the present error. A higher proportional gain results in a stronger response to the error, which helps reduce the steady-state error but may lead to overshoot and oscillations.

The integral component integrates the accumulated error over time and produces a control signal based on the integral of the error. It accounts for any steady-state error that persists over time. The integral action helps eliminate the offset between the setpoint and the actual speed, bringing the error to zero in the long run. However, too high of an integral gain can lead to instability or oscillations[9].

Derivative factor helps anticipate and counteract the error trend. The derivative action can increase system stability, reduce overshoot, and improve response time. However, excessive derivative gain can amplify noise and cause instability[10].

By combining these three actions, the PID controller dynamically adjusts the motor's input voltage or current to maintain the desired speed. The controller continuously monitors the feedback signal (e.g., from an encoder or tachometer) and updates the control signal based on the calculated error and the gains associated with each component [11].

The tuning of the proportional Integral Derivative manager includes regulating the relational, essential, and copied advances to attain the wanted speed response characteristics, such as fast response, minimal overshoot, and low steady-state error. Several tuning methods, such as manual tuning, Ziegler-Nichols method, and auto-tuning algorithms, can be employed to optimize the PID controller's performance [12-14].

PID speed control of DC motors finds applications in various fields, including robotics, automation, motion control systems, and industrial processes. It enables precise speed regulation, improved motor performance, and enhanced system stability, making it a fundamental technique in motor control engineering.

A mathematical model of the DC motor is developed based on its electrical and mechanical characteristics. The model represents the motor's dynamic response to input signals, such as voltage or current. It typically includes equations that describe the motor's electrical equations, torque-speed characteristics, and mechanical dynamics [15].

Using the motor model, a model-based controller is designed to generate control signals based on the desired motor performance. The controller uses the model to predict the motor's behavior and calculates the control signals needed to achieve the desired response. The PID control algorithm is typically implemented as part of the model-based controller.

The PID control algorithm is integrated into the model-based controller. The controller continuously measures the motor's actual speed or position and compares it to the desired setpoint. The error between the actual and desired values is calculated [16].

The integral component integrates the accumulated error over time and multiplies it by an integral gain to produce a control signal. The integral action accounts for any persistent steady-state error and continuously adjusts the control signal to eliminate it over time [17].

The derivative component calculates the rate of change of the error and multiplies it by a derivative gain. The derivative action anticipates the error trend and generates a control signal that counteracts rapid changes in the error, improving system stability and response time.

The PID controller combines calculation to final control signal. The control signal is then applied to the DC motor's input, such as voltage or current, to adjust its speed or position.

The model-based controller continuously measures the motor's actual speed or position through feedback sensors, such as encoders or tachometers. The feedback signal is used to compute the error and adjust the control signal accordingly. This closed-loop feedback system ensures that the motor's performance is continuously monitored and corrected to maintain the desired speed or position.

By incorporating the motor model into the control strategy, the model-based controller can provide enhanced performance, improved stability, and better disturbance rejection. The PID control algorithm, in conjunction with the motor model, enables accurate and responsive control of the DC motor system, making it suitable for applications that require precise motion control and high-performance operation. A PID controller works well. They all come down to employing a model built into the controller. Just refer to them as "model-based controllers" for now. In order to get a somewhat better responsiveness than PID, model-based controllers must trade resilience [15,16]. PID controller design methods can be classified into two categories: time-domain and frequency-domain methods. These methods aim to achieve desired performance criteria, such as settling time, overshoot, or gain and phase margins, either in the time domain or frequency domain. Here are some commonly used PID controller design methods for each criterion Time-Domain Performance Criteria. Ziegler-Nichols Method based on step response analysis. Determines the critical gain and oscillation period to set the proportional and integral gains. Three variations: Ziegler-Nichols ultimate gain method, Ziegler-Nichols ultimate gain with overshoot, and Ziegler-Nichols quarter decay ratio method.

Cohen-Coon Method: Uses the step response of the system to estimate model parameters. Provides approximate gains for proportional and integral control. Suitable for systems with varying time constants.

Internal Model Control (IMC): Employs an internal model of the process to design the controller. Considers the desired closed-loop response and model dynamics. Suitable for processes with time delay.

Direct Synthesis: Designs the PID controller directly from desired closed-loop transfer function or step response specifications. Requires knowledge of the system model or desired response.

Frequency-Domain Performance Criteria.

Gain and Phase Margin Specifications: Sets desired gain and phase margins to ensure stability and robustness. Analyzes the open-loop transfer function or Bode plot of the system. Adjusts PID gains to achieve the desired margins.

Loop Shaping: Modifies the open-loop transfer function to achieve desired frequency response characteristics.

Shapes the magnitude and phase response of the open-loop system using filters or compensators. Requires knowledge of the system dynamics and desired performance. Internal Model Control with Frequency Domain Specifications: Combines frequency-domain analysis with internal model control. Considers desired frequency response specifications, such as bandwidth or resonant frequency. Adjusts the PID gains based on the internal model and desired response.

It's important to note that different design methods have their strengths and limitations, and the selection of a specific method depends on the specific requirements of the control system, the available system information, and the desired performance criteria. Additionally, tuning the PID controller often involves an iterative process of adjusting the gains and evaluating the system's response to achieve the desired performance.

Model Predictive Control (MPC) is a widely used advanced control technique that enables precise control of complex dynamical systems. MPC utilizes a predictive model of the system to optimize control actions over a finite time horizon, taking into account future system behavior and desired performance criteria [18-19]. One popular variant of MPC is the integration of Proportional-Integral-Derivative (PID) control technique. PID control is a classic and well-established control strategy known for its simplicity and robustness. By combining the predictive capabilities of MPC with the inherent stability of PID control, an effective control approach can be achieved, allowing for superior performance in a wide range of applications. This article explores the principles and benefits of using PID control in model predictive control and highlights its application in various fields.

The MPC framework consists of three main components: the prediction model, the optimization problem formulation, and the control law. The optimization problem formulates the objective and constraints, aiming sequence. Finally, the control law computes the control signal based on the optimization results.

Robustness: PID control is known for its robustness and stability properties. By integrating PID within MPC, the resulting control system inherits these advantages, ensuring effective control even in the presence of disturbances or modeling uncertainties. Simplicity: PID controllers are simple to implement and tune. This simplicity makes them suitable for real-time applications and enables easy integration into the MPC framework

Jin et al. [11] offered three tuning guidelines for a straightforward but efficient functioning. The proportional gain and differential time utilized in this method are identical to those in a linear controller because it conditional methodology is working significantly sound. As a result, only the integral action is needed to change the error's magnitude. The simulation outcomes demonstrated that the suggested approach outperformed Tavakoli's PID controller [13].

One of the researcher [21] presented a method called Control-system synthesis. However, the strongly stabilising requirements for the time delayed unstable processes are still not obvious in the literature. Direct synthesis and disturbance rejection were the foundations for the PI/PID controller architecture that D. Chen and D.E. Seborg [13] suggested, although it was only effective for stable systems[14], although the reaction of such systems is quite slow.

Due to its broad, straightforward, and continuous control capabilities, DC motors have been extensively used in a variety of industrial applications, including electric cranes, electric vehicles, and robotic manipulators [16].

A. DC MOTOR

DC motors are capable of speed control, which allows them to modify their speed, torque, and even rotational direction at any time to adapt to changing circumstances [20]. The accompanying figures 1, and 2 depict the rotor's free body diagram as well as the armature's electrical circuit. It finds applications in various industries, ranging from manufacturing and robotics to automotive and aerospace. One of the key requirements in many of these applications is the ability to control the speed of the DC motor. In this explanation, we will delve into the reasons why DC motors need speed control, the benefits it offers, and the different methods employed to achieve speed control. Speed control is closely related to torque control in DC motors. Torque is the rotational force generated by the motor, and it is directly proportional to the current flowing through the motor's windings. By controlling the motor speed, the torque can be adjusted to match the requirements of the application. This allows for precise control over the motor's output, ensuring that it provides sufficient torque for tasks while avoiding excessive or insufficient force. Torque control is vital in applications such as robotics, where precise force control is necessary for delicate tasks.

Three parameters must be specified in order to create such a controller: proportional gain, integral time constant, and derivative time constant. The creation of techniques to shorten the time spent selecting the best controller parameter settings has received a lot of attention to date [21].

Speed control plays a critical role in maintaining system stability and ensuring a rapid response to changes in load or operating conditions. In many applications, sudden changes in load or speed can occur due to variations in the process or external factors. Without speed control, the motor may not be able to respond quickly enough, leading to instability, reduced performance, and even damage to the motor or the driven system. By employing speed control mechanisms, the motor's response time can be optimized, allowing it to adapt to dynamic conditions and maintain stable operation.

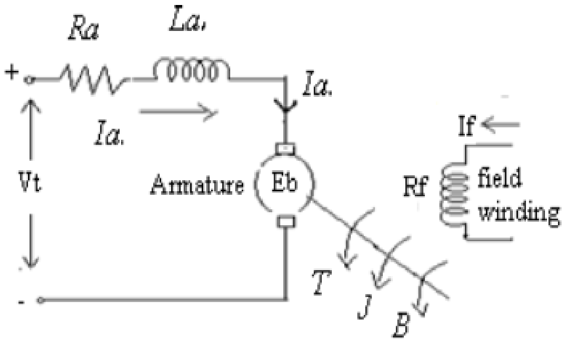
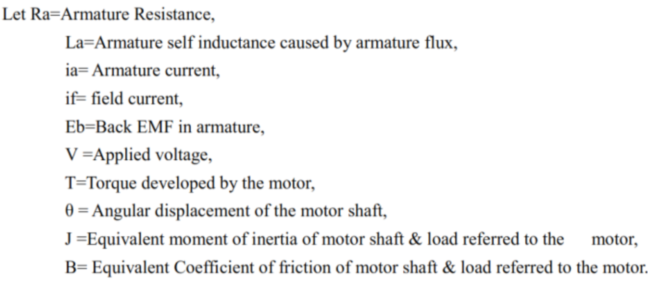


Fig.1. DC MOTOR



Let Ra=Armature Resistance,

La=Armature self inductance caused by armature flux,

ia= Armature current,

if= field current,

Eb=Back EMF in armature,

V =Applied voltage,

T=Torque developed by the motor,

θ = Angular displacement of the motor shaft,

J =Equivalent moment of inertia of motor shaft & load referred to the motor,

B= Equivalent Coefficient of friction of motor shaft & load referred to the motor.

B. PID Controller

It is represented by a gain constant (Kp) that scales the error value. The proportional control output is directly proportional to the error signal. Increasing the proportional gain amplifies the control action's effect, while reducing it weakens the control action. PID control action refers to the fusion of proportional, integral, and derivative control actions. Many different kinds of dynamic plants can have their time-domain behaviour controlled by PID controllers. [22]. It accounts for the accumulated past errors and generates an output that is proportional to the integral of the error. The integral control block is represented by a gain constant (Ki). Integrating the error helps to reduce the offset between the setpoint and the process variable.

This error signal is essential for the PID controller to assess the system's current state and determine the necessary corrective action. The error calculation is represented by the subtraction of the setpoint from the process variable.

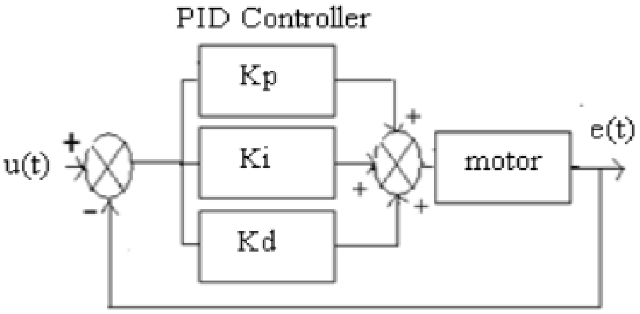


Fig.2. DC MOTOR with PID

By comparing the process variable with the setpoint, the feedback path provides information on how the system is responding to the control action. This feedback enables the controller to continuously adjust its output to minimize the error and bring the system closer to the desired setpoint. The block diagram components mentioned above work together to create a closed-loop control system. The error calculation generates the error signal, which is processed by the proportional, integral, and derivative control blocks. The control action output is then applied to the plant, and the resulting process variable is fed back to the error calculation block.

$$u = kpe + ki \int edt + kd \frac{de}{dt} \quad [1]$$

The summing junctions are represented by "+" symbols in the block diagram. They combine the outputs of the proportional, integral, and derivative control blocks. The summing junction after the error calculation block combines the error signal with the outputs from the integral and derivative control blocks. The resulting sum represents the total control action that needs to be applied to the system.

The plant or process block represents the system being controlled. It could be a physical process, such as a motor, a chemical reactor, or a temperature control system. The control variable, adjusted by the PID controller, influences the plant's behavior, and the output of the plant feeds back to the error calculation block.

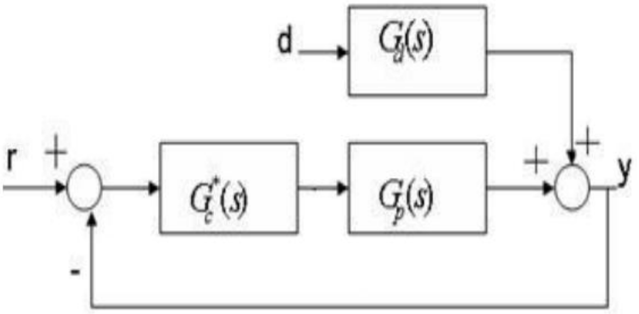


Fig.3.Classical Feedback control system

Consider of a response regulator scheme along with the typical Figure 3 wedge illustration. Let that the process, measurement component, transmitter, and control valve are all models in Gp(s). The following equation derives the close loop constraint for alters: The reference speed plays a vital role in the I term's operation. As the motor operates, the error accumulates and integrates over time. This accumulation is based on the deviation of the actual speed from the reference speed. A higher reference speed implies a larger deviation, resulting in a larger accumulated error. Consequently, the I term contribution to the control signal increases, allowing the system to eliminate steady-state error more effectively.

The reference speed comes into play by providing the basis for comparison with the actual speed. The P term calculates the error by subtracting the actual speed from the reference speed. A higher reference speed will lead to a larger error, which, in turn, will result in a more substantial contribution from the P term to the control signal. This helps in achieving faster response and reducing steady-state error for high reference speeds.

The reference speed, also known as the setpoint, is the target speed at which the DC motor is expected to operate. It is a user-defined parameter that represents the desired operating point for the motor. The choice of the reference speed depends on the specific application and the requirements of the system.

When designing a speed control system for a DC motor, the primary objective is to maintain the motor's speed as close as possible to the reference speed. This is achieved by employing a PID (Proportional-Integral-Derivative) controller, which is a widely used feedback control algorithm.

$$\frac{y}{r} = \frac{Gp(s)Gc(s)}{1 + Gp(s)Gc(s)} \quad [2]$$

The tachometer continuously measures the actual speed of the motor and provides feedback to the system. This feedback signal is used to update the error calculation and adjust the control signal through the PI controller.

The feedback loop in this classical control system ensures that the actual speed of the DC motor closely follows the desired speed set by the setpoint. By continuously monitoring the speed and making adjustments based on the error, the system achieves speed control and maintains stability.

By incorporating the reference speed and utilizing the PID controller, the speed control system can accurately regulate the motor's speed, minimizing deviations from the desired operating point. The PID controller continuously adjusts the control signal based on the error, allowing the motor to maintain stability and achieve the reference speed effectively. Although the reference speed does not have a direct impact on the D term, it indirectly influences the system's response to changes in the error. A significant change in the reference speed can result in a rapid change in the error signal. The D term reacts to this rate of change, contributing to the control signal accordingly. Therefore, a higher reference speed may lead to more pronounced adjustments in the motor's speed to counteract sudden changes in the error.

$$Gc(s) = \frac{\left(\frac{y}{r}\right)}{Gp(s)\left[1 - \frac{y}{r}\right]} \qquad [3]$$

II. PROPOSED METHODOLOGY

To control speed of DC motor using Modified IMC direct synthesis is proposed in this paper. The mathematical modelling equation 4, 5 are used which used to derived the transfer function of dc motor. Figure 4 shows the equivalent circuit of dc motor.

$$Va(t) = Ra.ia + La.\frac{d\ ia\ (t)}{dt} + eb(t) \qquad [4]$$

$$Tm(t) = Jm.\frac{d\ \omega\ (t)}{dt} + Bm.\omega(t) + Tl(t) \qquad [5]$$

By using above Eq4 and Eq5 we get transfer function of motor

$$\frac{\omega(s)}{V(s)} = \frac{Kt}{(Las + Ra)(Jms + Bm) + KtKb} \qquad [6]$$

Where:

I_a= armature current (A)

V_a= armature voltage (V)

E_b= back-emf voltage (V)

K_b= back-emf constant (V/rad/s)

L_a= armature circuit inductance (H)

R_a= armature circuit resistance (Ω)

ω = motor speed (rad/s)

J_m= moment of inertia of (load and rotor) (N.m^2)

B_m = viscous friction constant (N.m/rad/s)

K_T= torque constant (N.m/A)

T_L= load torque (N-m)

T_m = motor developed torque (N-m)

$$\frac{y}{d} = \frac{Gd\ (s)}{1 + Gp\ (s)\ Gc\ (s)} \qquad [7]$$

Rearranging gives a feedback controller expression

$$Gc(s) = \frac{Gd\ (s)}{\left(\frac{y}{d}\right) Gp\ (s)} - \frac{1}{Gp(s)} \qquad [8]$$

$$G_c(s) = \frac{\tilde{G}_d(s)}{\left(\frac{y}{d}\right)_d \tilde{G}_p(s)} - \frac{1}{\tilde{G}_p(s)} \qquad [9]$$

$$\left(\frac{y}{d}\right)^{DS-d} = \frac{\tilde{G}_p G_d \left(\frac{y}{d}\right)_d}{G_p \tilde{G}_d + \left(\frac{y}{d}\right)_d (\tilde{G}_p - G_p)} \qquad [10]$$

The closed-loop transfer function is used in the ideal case where the model is :

$$\left(\frac{y}{d}\right)^{DS-d} = \left(\frac{y}{d}\right)_d \qquad [11]$$

$$\left(\frac{y}{r}\right)^{DS-d} = 1 - \frac{\left(\frac{y}{d}\right)_d}{\tilde{G}_d(s)} \qquad [12]$$

III. RESULT AND DISCUSSION

MATLABR2022 is the software version used to construct the proposed system simulation. The findings of the suggested methodology are discussed in this section.

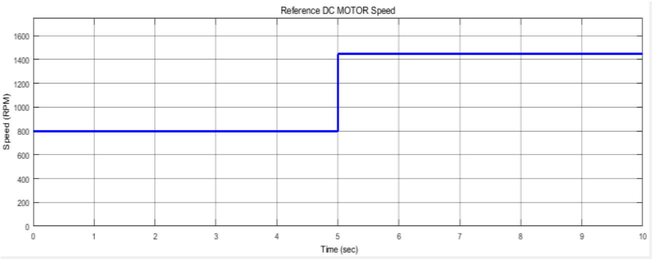


Fig.4. DC MOTOR Reference Speed

The above figure 4 is showing the reference speed which is given to controller. 800 rpm is from 0 to 5 sec then after 5 sec speed goes to 1440 rpm. Using proposed technique, figure 5 shows the speed curve. The speed curve is settling very quickly with respect to reference speed.

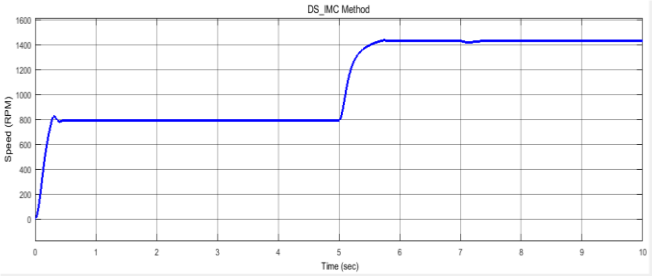


Fig.5. Speed curve using DS-IMC

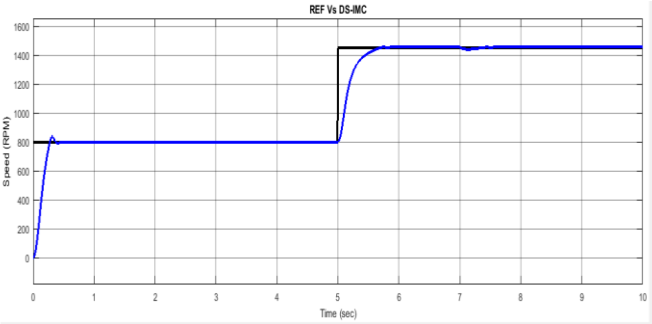


Fig.6. Ref Speed Vs DS-IMC

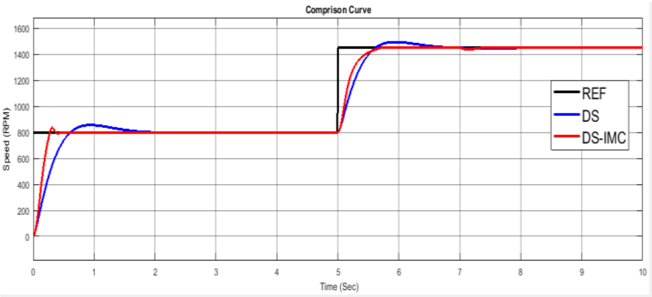


Fig.7. Comparative Results

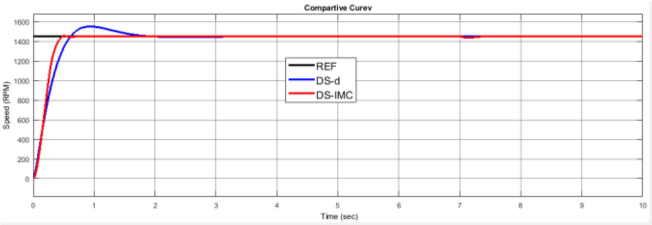


Fig.8. With no disturbance

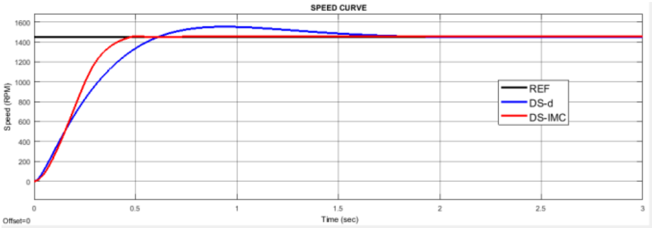


Fig.9. With no disturbance

The compartive anlysis is showing in figure 7 from which is it clear that proposed thechnique is having quit good resopnse as compared to DS-d method.when speed having constant nature , then comparsion between both technique is showing in figure 8.The figure 9 ,having less simulation time to show the transient part of all techniques. Its is understand by figure 9 , response is too good as compared to other technique.

IV. CONCLUSION

The primary goal of this project is to develop resilient PID controllers using the DS-d and IMC Modified approaches. The single design parameter in system approach is determined analytically using it. As a result, the DS-d design process work in significant functioning. Despite the fact that PID controllers are built to reject disturbances, set-point responses are typically sufficient. Different design approaches have been contrasted using two simulation instances.

REFERENCES

1. O'Dwyer, A. Handbook of PI and PID Controller Tuning Rules; Imperial College Press: London, UK, 2009.
2. Pathiran, A.R.; Prakash, J. Design and implementation of a model-based PI-like control scheme in a reset configuration for stable single-loop systems. Can. J. Chem. Eng. 2014, 92, 1651–1660.

3. Tavakoli, S.; Tavakoli, M. Optimal tuning of PID controller for first order plus time delay models using dimensional analysis. In Proceedings of the Fourth International Conference on Control and Automation, Montreal, QC, Canada, 12 June 2003;pp. 942–946.
4. Cvejn, J. Sub-optimal PID controller settings for FOPDT systems with long dead time. J. Process Control 2009, 19, 1486–1495.
5. Luyben, W.L. Design of proportional integral and derivative controllers for integrating dead-time processes. Ind. Eng. Chem. Res.1996, 35, 3480–3483.
6. Kookos, I.K.; Lygeros, A.I.; Arvanitis, K.G. On-line PI controller tuning for integrator/dead time processes. Eur. J. Control 1999, 35,19–31.
7. Chidambaram, M.; Sree, R.P. A simple method of tuning PID controllers for integrator/dead-time processes. Comput. Chem. Eng.2003, 27, 211–215.
8. Sree, R.P.; Chidambaram, M. A Simple and Robust Method of Tuning PID Controllers for Integrator/Dead Time Processes.Chem. Eng. Jpn. 2005, 38, 113–119.
9. Ali, A.; Majhi, S. Integral criteria for optimal tuning of PI/PID controllers for integrating processes. Asian J. Control 2010, 13,328–337.
10. Anil, C.; Sree, R.P. Design of optimal PID controllers for integrating systems. Indian Chem. Eng. 2014, 56, 215–228.
11. Jin, G.-G.; Son, Y.-D. Design of a Nonlinear PID Controller and Tuning Rules for First-Order Plus Time Delay Models. Stud.Inform. Control 2019, 28, 157–166.
12. Skogestad, S. Simple analytic rules for model reduction and PID controller tuning. J. Process Control 2003, 13, 291–309.
13. Jin, Q.; Liu, Q. Analytical IMC-PID design in terms of performance/robustness tradeoff for integrating processes: From 2-Dof to1-Dof. J. Process Control 2014, 24, 22–32.
14. Lee, J.; Cho, W.; Edgar, T.F. Simple Analytic PID Controller Tuning Rules Revisited. Ind. Eng. Chem. Res. 2013, 53, 5038–5047.
15. Shamsuzzoha, M. A unified approach for proportional-integral-derivative controller design for time delay processes. Korean J. Chem. Eng. 2015, 32, 583–596.
16. Ahmadi, A.H.; Nikravesh, S.K.; Amani, A.M. A Unified IMC based PI/PID controller tuning approach for time delay processes. AUT J. Electr. Eng. 2020, 52, 31–52.
17. Li, Z.; Bai, J.; Zou, H. Modified two-degree-of-freedom Smith predictive control for processes with time-delay. Meas. Control 2020, 53, 691–697. [CrossRef]
18. Chen, D.; Seborg, D.E. PI/PID Controller Design Based on Direct Synthesis and Disturbance Rejection. Ind. Eng. Chem. Res. 2002, 41, 4807–4822. [CrossRef]
19. Rao, A.S.; Rao, V.S.R.; Chidambaram, M. Direct synthesis-based controller design for integrating processes with time delay.
20. J. Frankl. Inst. 2009, 346, 38–56.
21. Ajmeri, M.; Ali, A. Direct synthesis based tuning of the parallel control structure for integrating systems. Int. J. Syst. Sci. 2015, 46, 2461–2473.
22. Anil, C.; Sree, R.P. Tuning of PID controllers for integrating systems using direct synthesis method. ISA Trans. 2015, 57, 211–219.
23. Yerolla, R.; Besta, C.S. Development of tuning free SISO PID controllers for First Order Plus Time Delay (FOPTD) and First OrderLag Plus Integral Plus Time Delay model (FOLIPD) systems based on partial model matching and experimental verification.Results Control Optim. 2021, 5, 100070.
24. Korkmaz, M.; Aydogdu, O.; Dogan, H. Design and performance comparison of variable parameter nonlinear PID controller and genetic algorithm based PID controller. In Proceedings of the 2012 International Symposium on Innovations in Intelligent Systems and Applications (INISTA 2012), Trabzon, Turkey, 2–4 June 2012; pp. 1–5.
25. So, G. Design of an Intelligent NPID Controller Based on Genetic Algorithm for Disturbance Rejection in Single IntegratingProcess with Time Delay. J. Mar. Sci. Eng. 2020, 9,
26. Curve Fitting Toolbox. In User's Guide; Mathworks: Natick, MA, USA, 2020.
27. System Identification Toolbox. In Getting Started Guide; Mathworks: Natick, MA, USA, 2020.
28. System Identification Toolbox. In User's Guide; Mathworks: Natick, MA, USA, 2020.
29. Md Akram Ahmad,Kamal Kishor,Pankaj Rai Speed Control of a DC Motor using Controllers: Automation Control and Intelligent Sydtem 2014; 2(6-1): 1-9