

# DC Motor Speed Control using Internal Model Controller: Industrial Transformation Strategy

Usman Omeiza Ahmed, Ayuba Avong Patrick, Benjamin Amough Kwembe



**Abstract:** In developed nations, industries are made to function at control engineering costs via the use of appropriate control schemes for dc motors. This paper introduces the role played by dc motors in industries thereby necessitating the analysis and performance validation of dc motor in Internal Model Control (IMC) scheme as against the Proportional–Integral–Derivative (PID) control schemes that is widely used in most industries. Theories on dc motor model, PID and IMC controller were detailed to paved the way for the methodical approach of getting specifications and transfer function for a typical dc motor (model RMCS-3011). Matlab/Simulink software was then used to tune the PID controller for the purpose of finding the values of PID gains that meets the design requirements to achieve best performance, thereby enabling the simulation of the PID controller. Using Matlab m-file environment, IMC controller transfer function was generated and simulated. The IMC controller transfer function aimed at achieving a unity gain that tracks the set-point was approximately realized. In the realization process, it was obvious that a filter is required. The aim of this work is to evaluate the performance of the IMC controller over PID controller. Simulated plots in Matlab-Simulink using the PID gains for the PID controller, and time constants and filter order for the IMC were presented. The quantitative results of the IMC method when compared with that of PID control provides a commendable performance. However, the performance in terms of rise time is small and preferred with the use of Matlab-Simulink tuned PID controller. Conclusively, IMC controller would be the preferred controller where the robustness and accuracy of the dc motor speed control counts more than faster response

**Keywords:** Controller, dc motor, industries, model, performance.

## I. INTRODUCTION

DC motors are used to generate rotational speed and linear position control industrial applications which may include electric cranes, steel rolling mills, electric vehicles and robotic manipulators etc. [1]. The achievable efficiency controlling the motor's speed and torque is critical and of which higher performance value is vital for mentioned industrial applications. This necessitate the need for higher performing dynamic control specifically targeting transient response and steady-state response of the motor. Proportional–Integral–Derivative (PID) is the most used among other control schemes over the years;

Revised Manuscript Received on May 15, 2020.

\* Correspondence Author

Usman Omeiza Ahmed\*, Department of Electrical and Electronics Engineering Technology, Federal Polytechnic Nasarawa, Nasarawa, Nigeria. E-mail: eurocly@gmail.com

Ayuba Avong Patrick, Department of Electrical and Electronics Engineering Technology, Federal Polytechnic Nasarawa, Nasarawa, Nigeria. E-mail: deavong12@gmail.com

Benjamin Amough Kwembe, Department of Electrical and Electronics Engineering Technology, Federal Polytechnic Nasarawa, Nasarawa, Nigeria. E-mail: benkwembe@gmail.com

© The Authors. Published by Blue Eyes Intelligence Engineering and Sciences Publication (BEIESP). This is an open access article under the CC BY-NC-ND license (<http://creativecommons.org/licenses/by-nc-nd/4.0/>)

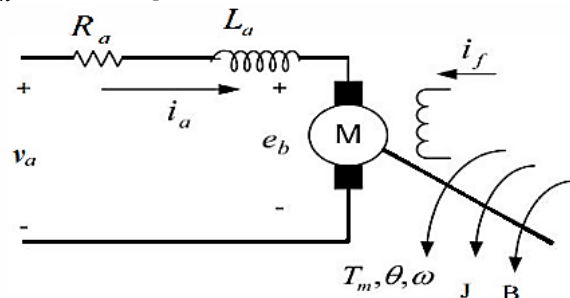
it enables flexible, precise and simplified speed and position control of the dc motor.

In this paper, the Internal Model Control (IMC) controller scheme was examined and verified using reference Matlab-tuned PID controller model. The identified inefficiencies of the conventional PID controller which is attribute of susceptibility to noise at high frequency, sensitivity to controller gain, undesirable speed overshoot at start and sluggish response due to sudden change in load torque [2]. Adopting PID controllers could be attributed to its unsatisfactory performance; reason been dc motors though modeled as linear systems are often subjected to nonlinearities. Therefore, implementing others control scheme could be optimization of system's stability and achieving satisfactory performance. This research's objective will include to validate performance of the IMC and PID controllers. DC motors are used in areas of applications where larger bandwidth speed is needed [2]. The ease in speed control gives dc motors greater advantage in applications where variable speed drives is required. Theoretical and real analysis have identified relationship between dc motor speed, armature voltage, magnetic flux per poles; Thus, speed of rotor can be set by controlling armature voltage and/or the field current.

## II. MODEL OF A DC MOTOR

Fig. 1 depicts the control equivalent circuit of a dc motor [3]. With reference to Fig. 1, the dc motor model can be described as follows:

- $R_a$  : armature resistance
- $L_a$  : armature inductance
- $i_a$  : armature current
- $i_f$  : field current
- $v_a$  : input voltage
- $e_b$  : back electromotive force (e. m. f)
- $T_m$  : motor torque



**Fig. 1: Control Equivalent Circuit of DC Motor [3]**

- $\theta$  : rotor angular speed
  - $\omega$  : rotor angular velocity
  - $J$  : moment of inertia
  - $b$  : viscous friction coefficient of the motor
- The armature voltage equation is given by

$$v_a = R_a i_a(t) + L_a \frac{d i_a(t)}{dt} + e_b(t) \quad (1)$$

The back e.m.f equation of the motor will be

$$e_b = K_b \omega(t) \quad (2)$$

where

$K_b$  : back e. m. f constant

From Newton's law, the motor torque can be obtained as

$$T_m(t) = J \frac{d^2 \theta(t)}{dt^2} + b \frac{d\theta}{dt} \quad (3)$$

$$T_m(t) = K_T i_a(t) \quad (4)$$

Thus,

$$K_T i_a(t) = J \frac{d^2 \theta(t)}{dt^2} + b \frac{d\theta}{dt} \quad (5)$$

$$K_T i_a(t) = J \frac{d\omega(t)}{dt} + b \omega(t) = T_m(t) \quad (6)$$

where

$K_T$  : Torque constant

Taking Laplace transform of equations (1), (2) and (6) respectively gives

$$V_a(s) = (R_a + L_a s) I_a(s) + E_b(s) \quad (7)$$

$$E_b(s) = K_b \omega(s) \quad (8)$$

$$K_T I_a(s) = J s \omega(s) + B \omega(s) = T_m(s) \quad (9)$$

Fig. 2 describes the dc motor armature control system function block diagram derived from equation (7) to (9). The input voltage  $V_a(s)$  and the output angular velocity  $\omega(s)$  are related by the transfer function:

$$G(s) = \frac{\omega(s)}{V_a(s)} = \frac{K_T}{(sL_a + R_a)(Js + B) + K_b K_T} \quad (10)$$

Equation (10), on expansion and factorization of its denominator can be rewrite as:

$$G(s) = \frac{\omega(s)}{V_a(s)} = \frac{K_T}{(L_a J)s^2 + (L_a B + R_a J)s + (R_a B + K_b K_T)} \quad (11)$$

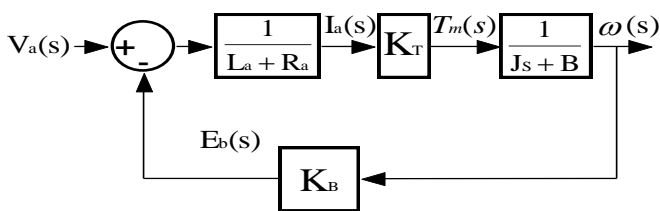


Fig. 2: Function Block Diagram of a DC Motor Armature Voltage Control System [3]

### III. SPEED CONTROL OF DC MOTOR

Electric motor convert electrical energy to mechanical energy by interactions of magnetic fields. One field is produced by a magnet of poles assembly; the other produced when an electrical current flowing in the motor windings. Advantages of dc motor include high start torque, high response performance, easier to be linear controlled and low cost etc. this makes them find widely applicable in the industries [4]. The characteristics equation that describes the relationship of speed of a dc motor is

$$N = \frac{V_a - I_a R_a}{K \Phi} \quad (12)$$

where

$N$  is the DC motor speed

$V_a$  is the armature terminal voltage

$I_a$  is armature current

$R_a$  is armature circuit resistance

$\Phi$  is the field flux

$K$  is armature constant and is given by

$$K = \frac{PZ}{2\pi a} \quad (13)$$

where

$P$  = No. of poles

$Z$  = Total no. of armature conductors

$a$  = No. of parallel path [5]

Equation (12) shows that for a dc motor there are basically three methods of speed control:

1. Variation of resistance in armature circuit.
2. Variation of field flux
3. Variation of armature terminal voltage.

### IV. CONTROL SCHEMES USED FOR SPEED CONTROL OF DC MOTOR

The load on a dc motor may vary over a range of speed depending on the type of application. Certain applications may demand high-speed control accuracy and good dynamic responses. Thus, the motor should be precisely controlled to give the desired performance. There are several controller types: proportional integral controller, PID controller, neural network controller, fuzzy logic controller and linear quadratic regulator controller [6].

#### A. Proportional-Integral-Derivative (PID) Controller

The PID controller is a control loop feedback mechanism that calculates an "error" value as the difference between a measured process variable and a desired set point; the error is minimized by adjusting the process control inputs [7]. A PID controller consists of parallel connections of three elements: proportional element, integral element and derivative element. These elements take the error as input, use it to compute new input process. Each new input tries level the measured process to the desired set point values. In addition, the PID controller is a design aimed at achieving optimal control performance by adjusting an appropriate proportional gain  $K_p$ , integral gain  $K_i$ , and differential gain  $K_d$  [3]. Fig. 3, shows a PID controller system block diagram. The mathematical expression of the PID controller is:

$$y(t) = K_p \left[ e(t) + \frac{1}{K_i} \int_0^t e(t) dt + K_d \frac{d}{dt} e(t) \right] \quad (14)$$

The Laplace domain representation of PID controller is:

$$\frac{Y(S)}{E(S)} = K_p \left[ 1 + \frac{1}{K_i S} + K_d S \right] \quad (15)$$

PID controllers have the following features:

- They have the capacity to eliminate steady-state error of the response to a step reference signal (because of integral action).
- They have the ability to anticipate output changes (when derivative action is employed).

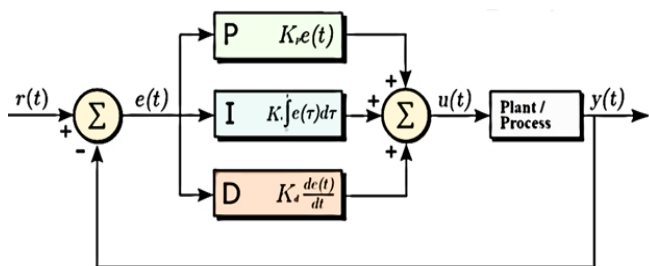


Fig. 3: PID Controller System Block Diagram [8]

**B. Internal Model Control (IMC) Controller**

Linear control schemes may not guaranty satisfactory performance when implemented upon a linear modeled system such as a dc motor [9]. This may be as a result of variation in motor-load dynamics and due to dc motor armature reaction introducing nonlinearities. The use of internal model control (IMC) controller can give better performance amidst external disturbances that risk the stability of a close loop system.

The theory of IMC states thus: “control can be achieved only if the control system encapsulates, either implicitly or explicitly, some representation of the process to be controlled” [10]. This encapsulation is obvious, as the IMC controller is designed using the inverse of the intended control process model. This is realized by cascading the IMC controller with the process transfer function, such that the process model is in parallel with the actual process. Thus, unity gain is achieved that results in an accurate tracking of the set-point. Fig. 4 is the block diagram of an IMC controller [9]. The block symbols and signals are defined as follows:

- Y(s): Output signal
- R(s): Set-point
- d(s): Disturbance signal
- $G_{IMC}(s)$ : Internal Model Control (IMC) Controller
- $G_P(s)$ : Actual process to be controlled
- $G_M(s)$ : Model of the actual process

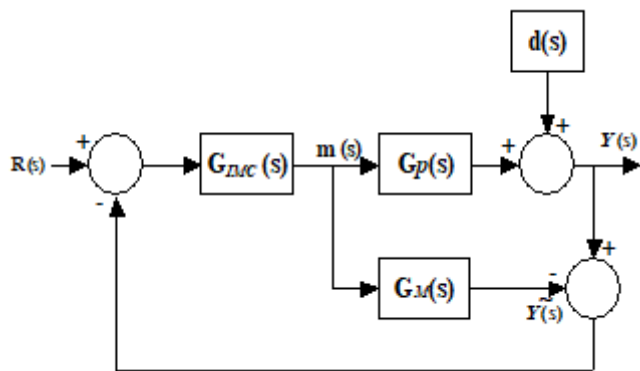


Fig. 4: Block Diagram of IMC [9]

For the purpose of simplification, the IMC block diagram of Fig. 4 can be redrawn as shown in Fig. 5. From Fig. 4,

$$Y(s) = G_P(s) m(s) + d(s) \tag{16}$$

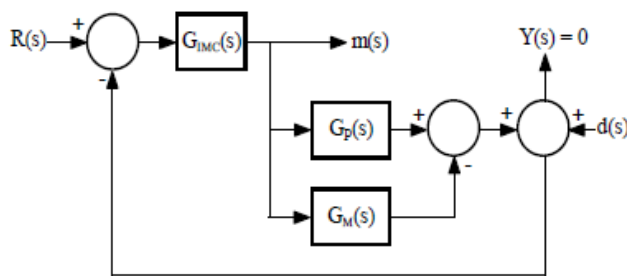


Fig. 5: Restructured Block Diagram of IMC

Using block diagram reduction technique, Fig. 5 is reduced to Fig. 6.

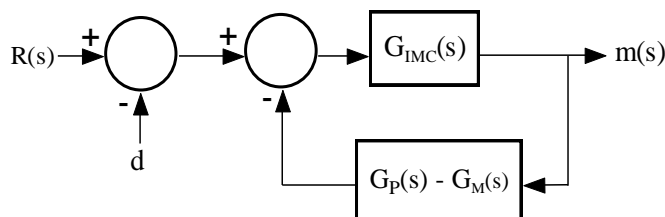


Fig. 6: Resultant IMC Equivalent Block Diagram

With respect to Figure 9 and letting  $Y(s) = 0$ .

$$m(s) = \frac{G_{IMC}(s)}{1 + G_{IMC}(s) [G_P(s) - G_M(s)]} [R(s) - d(s)] \tag{17}$$

Substituting equation (17) into equation (16) yields

$$Y(s) = d(s) + \frac{G_P(s) G_{IMC}(s)}{1 + G_{IMC}(s) [G_P(s) - G_M(s)]} [R(s) - d(s)] \tag{18}$$

“Perfect” control could be achieved by selecting

$$G_{IMC}(s) = \frac{1}{G_P(s)} \tag{19}$$

This can be verified by substituting equation (19) into equation (18) while applying the equality of the process transfer function to that of the process model, i.e.,  $G_P(s) = G_M(s)$  [11].

$$Y(s) = d(s) + \frac{G_P(s) G_{IMC}(s)}{1 + G_{IMC}(s) [G_P(s) - G_M(s)]} [R(s) - d(s)]$$

$$Y(s) = d(s) + \frac{G_P(s) \frac{1}{G_P(s)}}{1 + G_{IMC}(s) [G_M(s) - G_M(s)]} [R(s) - d(s)] \tag{20}$$

Equation (20) shows that the IMC whose transfer function is symbolized by  $G_{IMC}(s)$  can make the output,  $Y(s)$  to track the input,  $R(s)$ . Since  $G_P(s) = G_M(s)$ , equation (19) can be written as

$$G_{IMC}(s) = \frac{1}{G_M(s)} \tag{21}$$

In accordance with dual stability criterion, when the model is exact representation of the process, stability of both controller and plant (the actual process) is sufficient for overall system stability. That in itself depends on if the actual process is open loop stable. However, due to modelling error, the dual stability criterion may not be satisfied [11]. There is the need for the internal model control controller,  $G_{IMC}(s)$ , to be stable and realizable. Thus, a factorization of  $G_M(s)$  in equatio (21) is introduced such that

$$G_{IMC}(s) = \frac{1}{G_{MM}(s) G_{MA}(s)} \quad (22)$$

where,  $G_{MM}(s)$  is the transfer function component of  $G_M(s)$  that has minimum phase characteristics (both  $G_{MM}(s)$  and its inverse  $\frac{1}{G_{MM}(s)}$  are causal and stable- by not having zeros and poles on the right-half plane).  $G_{MA}(s)$  is the transfer function component of  $G_M(s)$  that has non-minimum phase (NMP) characteristics which is undesirable part largely responsible for the instability of the IMC controller. Thus, for a stable IMC, the approximation of equation (23) results having factored out  $\frac{1}{G_{MA}(s)}$ .

$$G_{IMC}(s) \cong \frac{1}{G_{MM}(s)} \quad (23)$$

$$G_{MM}(s) \cong \frac{1}{G_{IMC}(s)} \quad (24)$$

Practical IMC controller,  $G_{IMC,P}(s)$  that is stable and robust can be realized by factoring in (in place of the factored out unstable NMP component,  $\frac{1}{G_{MA}(s)}$ ) the transfer function of an nth-order low pass filter,  $G_F(s)$  to the IMC controller of equation (23). The filter is selected such that it has all roots inside the unit circle.

$$G_{IMC,P}(s) = G_{IMC}(s) G_F(s) = \frac{1}{G_{MM}(s)} G_F(s) \quad (25)$$

where,

$$G_F(s) = \frac{1}{(\lambda s + 1)^n} \quad (26)$$

where

$\lambda$  is the time constant

$n$  is the  $n^{\text{th}}$  order of the filter

Fig. 7 depicts the block diagram for the implementation of the internal model control scheme. An arbitrary gain of 0.001 was inserted along the feedback path to reduced and subsequently eliminate the external disturbances signal interference.

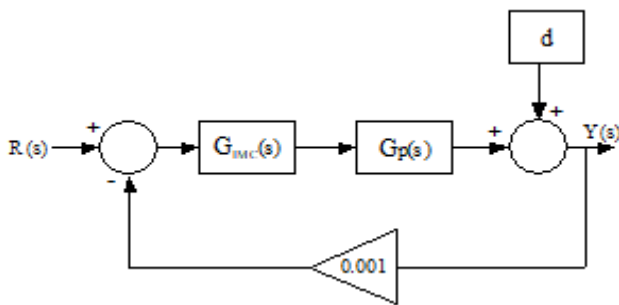


Fig. 7: Implementation Block Diagram for IMC Scheme

### V. CONTROL SYSTEM PERFORMANCE REQUIREMENT

Performance of a control system is often measured by applying a step function as the set point command variable, and then measuring the response of the process variable. Commonly, as illustrated in Fig. 8, the response is quantified by measuring defined waveform characteristics [12].

**Rise Time:** is the amount of time the system takes to go from 10% to 90% of the steady-state, or final, value.

**Percent Overshoot:** is the amount that the process variable overshoots the final value, expressed as a percentage of the final value.

**Settling time:** is the time required for the process variable to settle to within a certain percentage (commonly 5%) of the final value.

**Steady-State Error:** is the final difference between the process variable and set point.

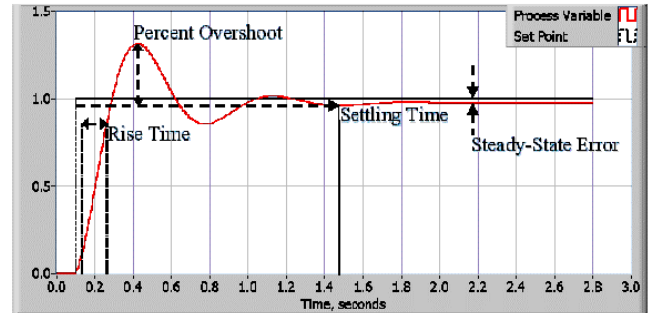


Fig. 8: Response of a Typical PID Closed Loop System [12]

### VI. METHOD

The methodology used in this research is outline sequentially: Determining dc motor (RMCS-3011) specifications; deriving dc motor (RMCS-3011) transfer function; tuning PID controller in Matlab-Simulink; simulation of PID controlled system; generating IMC controller transfer function, using Matlab; simulation of IMC Controlled System

#### A. DC Motor Specifications

The dc motor of choice is a geared brushed dc motor model RMCS-3011 with the following relevant manufacturer information:

Rated Voltage, $V_a$	:	12 V
Rated Torque, $T$	:	11.76798 Nm
Rated Speed, $N$	:	10 rpm
Torque Constant, $K_T$	:	66.4935 Nm A <sup>-1</sup>
Back e.m.f constant, $K_b$	:	18.2 mVs rad <sup>-1</sup>
Moment of Inertia, $J$	:	0.224 Kg m <sup>2</sup>
Motor armature resistance, $R_a$ :		2.96 $\Omega$
Motor armature inductance, $L_a$ :		2.51 mH

#### B. DC Motor (RMCS-3011) Transfer Function

Substituting the relevant manufacturer information and measured data of geared dc motor (model RMCS-3011) into equation (11) yields.

$$G_M(s) = \frac{1}{0.000008422s^2 + 0.010394s + 30.51677} \quad (27)$$

#### C. Simulation and Tuning of PID Controller

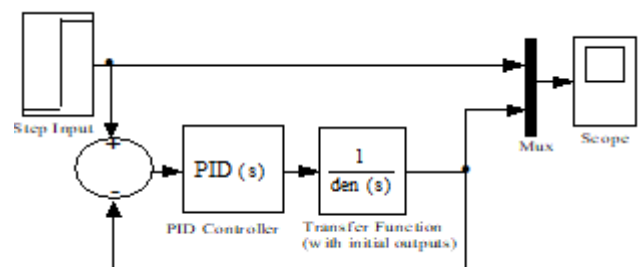


Fig.9: PID Controller Tuning Simulated Modelled Diagram for a DC Motor

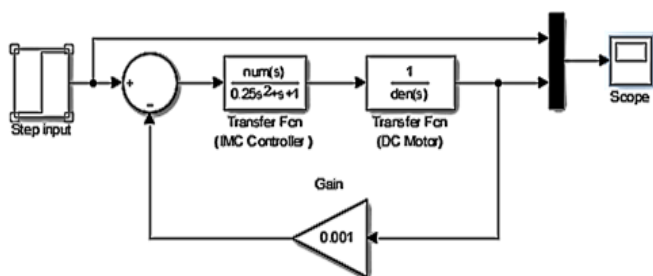


**D. Transfer Function of Practical IMC Controller**

Considering the transfer functions  $G_M(s)$  of equation (11) as the model of the process (i.e. the dc motor),  $G_F(s)$  of equation (26) as the model of nth order filter, equations (23), (24) and (25). With reference to the derivation of the IMC formula and derivations of equation (25) and (26), a program to generate transfer function for practical IMC controller,  $G_{IMC\_p}(s)$ , using Matlab m-file environment is thus:

```
% Program to generate transfer function of practical
internal %model controller, gIMC_P
% The numerator and denominator of the process (dc
motor) %model
numM = [1]; denM = [0.000008422 0.010394 30.51677];
gM = tf(numM,denM);
% Computing the transfer function of the inverse of the
%process model
gM_inv = inv(gM);
% Enabling transfer function to be entered in the 's' domain
S = tf('s');
% Generate an 'n' order filter (n = 1) transfer function
whilst %setting time constant, Y = 'certain variable value'
Y = 0.1; n = 1; gF = 1/((Y*S + 1)^n);
% Generating IMC transfer function
gIMC = gM_inv*gF;
% The transfer function that has minimum phase
%characteristics, gMM
gMM = inv(gIMC);
% Generating the Practical Internal Model Control
%Controller, gIMC_P
gIMC_P = gF/gMM
```

Within the program, the time constant,  $Y$ , can be varied whilst keeping the order of the filter,  $n = 1$ , constant. The numerator and denominator values of the generated transfer function,  $G_{IMC\_P}(s)$ , are entered into the IMC block of Simulink modeled IMC dc motor diagram (Fig. 10). The Scope block gives the performance response of the IMC.

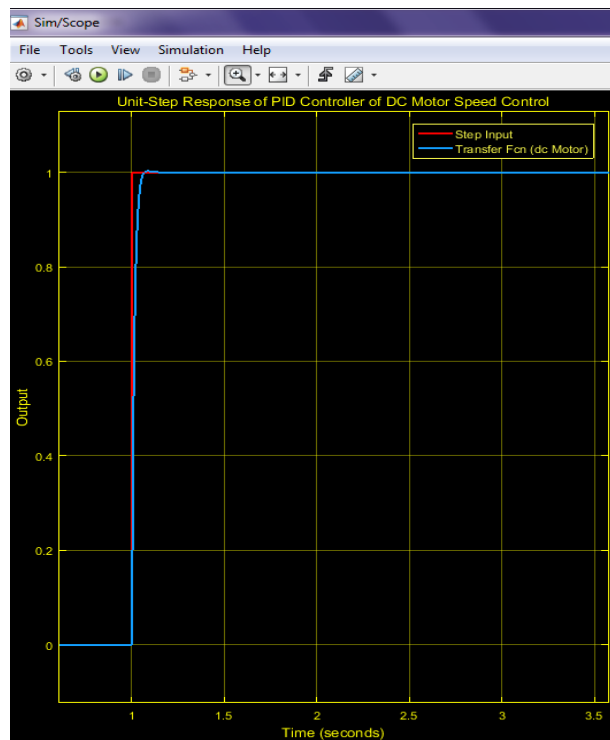


**Fig.10: Simulink Modelled IMC DC Motor Diagram**

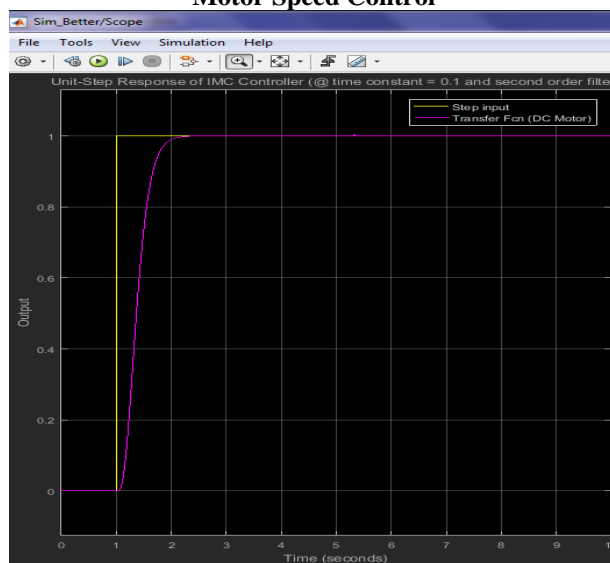
**VII. RESULTS**

The unit step response for tuned PID controlled dc motor speed control was plotted (Fig. 11). The unit-step response plots were generated for IMC controlled dc motor speed control at three varying time constant, ( $\lambda = 0.1$ ), ( $\lambda = 0.5$ ) and ( $\lambda = 0.9$ ) for both first order filter ( $n = 1$ ) and second order filter ( $n = 2$ ). Though, the best plot having  $\lambda =$

0.1 with  $n = 2$  is shown in Fig. 12. The performance parameters of each IMC plot were recorded and tabulated in Table-I. Lastly, the performance parameters of both controllers based on their respective plots were tabulated (Table-II) for the purpose of verification.



**Fig.11: Step Response of Tuned PID Controlled DC Motor Speed Control**



**Fig. 12: Step Response of IMC Controlled DC Motor Speed Control (Time Constant,  $\lambda = 0.1$ ;  $n = 2$ )**

Table – I: Comparison of IMC Controller Response

Parameters	IMC Controller					
	( $\lambda = 0.9; n = 1$ )	( $\lambda = 0.9; n = 2$ )	( $\lambda = 0.5; n = 1$ )	( $\lambda = 0.5; n = 2$ )	( $\lambda = 0.1; n = 1$ )	( $\lambda = 0.1; n = 2$ )
Overshoot (%)	Nil	Nil	0.505	Nil	0.437	0.026
Rise time (s)	2.966	4.247	1.649	2.425	0.330	0.492
Steady-state value	0.9900	0.9801	0.9904	0.9903	0.9909	0.9952
Steady-state error	0.0100	0.0199	0.0096	0.0097	0.0091	0.0048

Table-II: Comparison of Tuned PID Controller with IMC Controller

Parameter	Tuned PID Controller	IMC Controller ( $\lambda = 0.1; n = 2$ )
Overshoot (%)	0.505	0.026
Rise time (s)	0.0289	0.492
Steady-state value	0.9925	0.9952
Steady-state error	0.0075	0.0048

VIII. DISCUSSION OF RESULTS

The Matlab – Simulink tuned PID controller response of Fig. 11 is characterized by fast response as evident from small rise time. This desirable output is marred by the high overshoot which renders the PID controller to slightly unstable at start. In the case of the IMC controller, it is observed (from Table-I) that the best response (characterized by having very low overshoot and the least steady-state error) is achievable using a second order filter at time constant of 0.1.

From Table-II, the best response of IMC controller is noted to be have low steady-state error and overshoot when compared to that of Matlab-Simulink tuned PID controller. With reference to Table-I and Fig. 12, it is obvious that the overshoot is reduced to a large extent with IMC controller employed. However, the rise time is generally high for most varied time constant with any order of filter. The exception to this general trend is at time constant 0.1. In addition, as evident from Table-II, IMC provides better set point tracking than PID controller.

IX. CONCLUSION

In this research, the PID controller inarguably provides desirable performance but has undesirable demerits. The IMC controller was studied into for observable features to counteract these disadvantages. This research has successfully achieved its set objective and the conclusion (within the scope of the study) is as follows:

1. Comparatively, percentage overshoot with IMC controller is more desirably less than that of tuned-Matlab PID controller. The IMC Controller shows consistency in very low overshoot for varying time constant and order of filter.
2. Steady-state values for both IMC and tuned-Matlab PID controller are about equal. However, the IMC controller has a value much closer to the set-point.

Therefore, IMC controller would be the preferred controller where the robustness and accuracy of the dc motor speed control counts more than faster response.

REFERENCES

1. S. K. and S. D. K., "A Review of Transient and Steady State Response of DC Motor Position Control," *International Journal of Engineering Research & Technology*, vol. IV, no. 6, pp. 756-762, 2015.
2. S. A. Nisa, L. Mathew and S. Chatterji, "Comparative Analysis of Speed Control of DC Motor using AI Technique," *International Journal of Engineering Research and Applications*, vol. III, no. 3, pp. 1137-1146, 2013.
3. P. M. Meshram and R. G. Kanojiya, "Tuning of PID controller using Ziegler -Nichols method for speed control of DC motor," in *IEEE- International Conference On Advances In Engineering, Science And Management*, 2012.
4. M. Kushwah and A. Patra, "Tuning PID controller for speed control of dc motor using soft computing techniques-a review," *Advance in Electronic and Electric Engineering*, vol. IV, no. 2, pp. 141-148, 2014.
5. A. P. Singh, U. Narayan and A. Verma, "Speed Control of DC Motor using PID Controller based on Matlab," *Innovative Systems Design and Engineering*, vol. IV, no. 6, pp. 22-28, 2013.
6. N. Tripathi, R. Singh and R. Yadav, "Analysis of Speed Control of DC Motor – A Review Study," *International Research Journal of Engineering and Technology*, vol. II, no. 8, pp. 1616-1621, 2015.
7. S. Y. Moradi and E. Saedi, "Controlling DC Motor Position, Using PID Controller Made by PIC Microcontroller," *ZANCO Journal of Pure and Applied Sciences*, vol. XXVIII, no. 2, pp. 82-89, 2016.
8. elprocus, "The Working Principle of a PID Controller for Beginners," 2020. [Online]. Available: <https://www.elprocus.com/the-working-of-a-pid-controller/>. [Accessed 6 May 2020].
9. S. M. and P. D., "A Real Time IMC Tuned PID Controller for DC Motor," *International Journal of Recent Technology and Engineering*, vol. I, no. 1, pp. 65-70, 2012.
10. M. A. Ahmad, K. Kishor and P. Rai, "Speed Control of a DC Motor using Controllers," *Automation, Control and Intelligent Systems*, vol. II, no. 6, pp. 1-9, 2014.
11. C. E. Garcia and M. Morari, "Internal Model Control. A Unifying Review and Some New Results," *Industrial & Engineering Chemistry Process Design and Development*, vol. XXI, no. 2, pp. 308-323, 1982
12. National Instrument, "PID Theory Explained: Innovations," 17 March 2020. [Online]. Available: <https://www.ni.com/en-us/innovations/white-papers/06/pid-theory-explained.html>. [Accessed 6 May 2020].

AUTHORS PROFILE



Usman Omeiza Ahmed, is working as a Lecturer, Department of Electrical and Electronics Engineering Technology, Federal Polytechnic Nasarawa, Nasarawa, Nigeria. A registered member of the Council for the Regulation of Engineering in Nigeria (COREN), in addition to holding a corporate membership of the Nigerian Society of Engineers (NSE). He is also pursuing specialization in Control

Systems.





**Ayuba Avong Patrick**, is a holder of MS.c. with specialty in embedded systems applications designs. A Lecturer in the Department of Electrical and Electronic Engineering Technology,

Federal Polytechnic Nasarawa, Nasarawa, Nigeria and a member of Nigerian Society of Engineers (NSE).



**Kwembe Amough Benjamin**, is a lecturer in the Federal Polytechnic Nasarawa, Nigeria. He obtained his first degree, Bachelor in Communication Engineering from the University of Jos, Jos and Master of Engineering in Electronics from Abubakar Tafawa Balewa University Bauchi, Bauchi all in Nigeria. Kwembe has research interest in artificial in intelligence. He wrote a book titled "Fundamentals of Artificial Intelligence and

Robotics." His areas of research in the recent times include renewable energy and motor control systems.