Turkish Journal of Electrical Engineering & Computer Sciences

http://journals.tubitak.gov.tr/elektrik/

Research Article

Turk J Elec Eng & Comp Sci
(2017) 25: 4131 – 4141

doi:10.3906/elk-1611-311

# EKF-based self-regulation of an adaptive nonlinear PI speed controller for a DC motor

**Omer SALEEM*, Urwa OMER**
Department of Electrical Engineering, National University of Computer and Emerging Sciences, Lahore, Pakistan

**Received:** 30.11.2016 • **Accepted/Published Online:** 20.03.2017 • **Final Version:** 05.10.2017

**Abstract:** This paper presents a robust adaptive nonlinear proportional–integral (ANPI) scheme to control the speed of a direct-current (DC) motor. Unlike proportional–integral–derivative (PID) controllers, PI controllers have a simpler structure and they deliver effective control effort. However, due to inadequate controller gains, they are often unable to simultaneously improve the transient as well as the steady-state response of the system. A nonlinear PI (NPI) controller alleviates these issues and delivers a good response. In this research, the proportional and integral gains of the NPI controller are dynamically modulated via a nonlinear sigmoidal function (SiF) of the error dynamics of the motor's angular speed. The variation rates of these functions are manually tuned via trial-and-error method. These rates are also dynamically updated via an extended Kalman filter (EKF)-based adaptation mechanism. The performances of a linear PI controller, an NPI controller having fixed variation rates, and an NPI controller equipped with the EKF-based self-regulated SiFs are tested and compared in real time. The experimental results are analyzed to validate the effectiveness of the proposed ANPI controller in optimizing DC motor speed control.

**Key words:** DC motor, NPI control, sigmoidal function, self-tuning controllers, EKF

## 1. Introduction

Permanent magnet direct-current (DC) motors exhibit good servo performance, which makes them a popular choice as actuators in various household, automotive, and robotic applications [1]. High-precision machines require DC motors to demonstrate superior error-tracking capability and susceptibility to noise. Therefore, they must be equipped with robust speed controllers that are capable of adapting to random and abrupt changes in system's states. Conventional linear proportional–integral–derivative (PID) controllers are widely used in the process control industry due to their simple structure and ability to provide reliable control effort [2]. PI controllers are easier to implement in practice than PID controllers for speed control applications [3–5]. This is because the rate-of-change of error in speed is often a noisy signal. However, due to insufficient controller gains, PI controllers cannot simultaneously improve the steady-state and the transient response. Furthermore, fixed-gain PI controllers lack efficiency and robustness when confronted with complex electromechanical systems and systems with exogenous disturbances. Their capability to track the errors degrades significantly if the controlled system undergoes large input speed variations. Fuzzy logic-based self-tuning controllers are found to be quite effective [6–8]. However, the heuristically developed fuzzy rule-base(s) and membership functions cause several uncertainties in the system. Nonlinear PI controllers offer a viable solution to overcome the above-mentioned drawbacks by modulating the controller gains via a nonlinear function of error [9,10]. Several options are

*Correspondence: omer.saleem@nu.edu.pk

available for the choice of nonlinear functions [11]. A smooth sigmoidal function (SiF) is chosen in this research to adaptively tune the controller gains [12].

A typical NPI controller, with nonlinear gain functions having fixed variation rates, lacks robustness when faced with rapidly varying perturbations. Thus, an effective adaptation mechanism is required to self-regulate the variation rates and alleviate the weakness. These mechanisms include the derivative-free as well as the derivative-based methods [13]. The derivative-free methods do not depend on the derivative of the objective function and they converge to find the global minimum. However, their convergence rate is very slow. On the other hand, the derivative-based methods converge to find the local minimums at a faster convergence rate. Due to its faster convergence, a derivative-based method is adopted in this research for adaptive tuning of variation rate. The gradient-descent method is a widely used derivative-based autotuning technique. However, pertaining to its first-order characteristics, it converges slowly and cannot deliver good solutions [14]. The extended Kalman filter (EKF) is also an efficient and effective derivative-based optimization method. It can provide the best estimate of the variation rates in each step because it considers additional information (the actual estimates and their variances) regarding the system's state and error dynamics from the current and previous iterations [15,16]. Therefore, an EKF mechanism is investigated in this paper for the adaptive tuning of variation rates of an NPI motor speed controller.

## 2. Experimental setup

The proposed control law is tested on a QNET DC-Motor Control Trainer and NI-ELVIS II board [17,18]. It is shown in Figure 1. The trainer is integrated with LABVIEW software to visualize the motor's response [19]. The trainer comprises a permanent magnet DC motor that operates at $\pm 24$ V and 5 A, and has a 0.05 kg disc mounted at its shaft. Among other sensors, it contains a tachometer to measure the motor's angular speed ($\omega_r$). The moment of inertia ($J_{eq}$) of the motor armature is $9.46 \times 10^{-6}$ kg m$^2$, and its maximum torque is 0.14 Nm [20]. Since the motor has a small inertia, inductance, and torque, its mathematical model is approximated as first order, instead of the typical second order [21], and is given by Eq. (1).

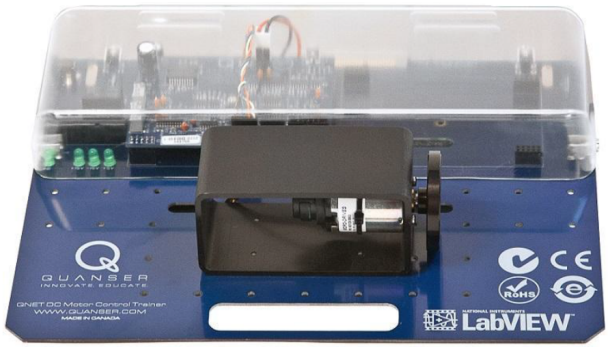


**Figure 1.** QNET DC motor control trainer.

$$\frac{\omega_r(s)}{V_m(s)} = \frac{K_m}{J_{eq}R_m s + K_m^2}, \tag{1}$$

where $V_m$ is the voltage signal applied to the motor, $K_m$ is the motor's back-emf constant (0.028 V/(rad/s)), and $R_m$ is the motor armature resistance (3.3 $\Omega$). The instantaneous angular speed of the motor ($\omega_r$) is acquired

via a tachometer and then serially transmitted (at 9600 bps) to the LABVIEW application on computer, where the proposed controller applies appropriate commands ($V_m$) to correct the motor response. The system is operated at a sampling frequency of 250 Hz.

## 3. Control system design

The proposed controller is used to efficiently track the reference speed of the motor, even in the presence of bounded exogenous disturbances. It minimizes the error in the motor's angular speed ($e_\omega$), given by Eq. (2). Its nonlinear nature improves the transient response by enhancing the error convergence rate and suppressing the overshoot and oscillation in the system. Initially the motor is tested with a linear PI controller. The proportional and integral gains are hard coded, at 0.085 and 1.65, respectively, via trial-and-error tuning method. The motor is then tested by introducing the SiFs in the controller to dynamically modulate its PI gains based on the changes in $e_\omega$ and its time-derivative ($\dot{e}_\omega$). In this case, the trial-and-error method is again used to fix the variation-rates of the SiFs. Finally, these variation-rates are self-regulated via the EKF. This enhancement transforms the controller into the proposed ANPI controller. The control law is given by Eq. (3).

$$e_\omega(t) = \omega_d(t) - \omega_r(t) \tag{2}$$

$$u(t) = K_p(e_\omega, \dot{e}_\omega) \cdot e_\omega(t) + \int_0^t [K_i(e_\omega, \dot{e}_\omega) \cdot e_\omega(t)]\, dt, \tag{3}$$

where $\omega_d(t)$ is the reference speed in rad/s, $\omega_r(t)$ is the actual motor speed in rad/s, and $K_p$ and $K_i$ are the SiFs to update the proportional and integral gain, respectively. The architecture of the proposed ANPI controller is shown in Figure 2.

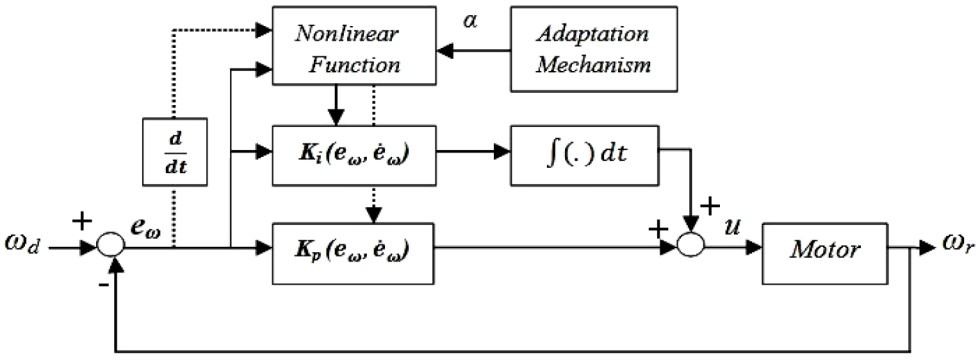


**Figure 2.** The proposed control architecture.

## 4. Nonlinear gain function

In a closed-loop system, a linear PI controller can either converge slowly (and monotonically) towards the desired output without any apparent overshoot, or it can exhibit a faster convergence with considerable overshoot and oscillations. Hence, the generic PI controllers are equipped with nonlinear gain functions so they can have a negligible overshoot and a faster convergence, simultaneously. Several options are available to be used as a bounded nonlinear gain functions of error dynamics to adaptively update the PI gains, such as the sigmoidal, hyperbolic, Gaussian, and the piece-wise linear function [22,23].

The sigmoidal function is preferred because it increases monotonically and is easily differentiable. Initially, when $e_\omega$ is large, the NPI controller delivers a large control command to reduce the rise-time so that the motor could reach the desired speed quickly. The SiFs continue to update the PI gains to optimize the correctional control effort as $\omega_r$ converges toward the reference value. This effect decreases the residual energy, increases the motors damping against oscillations, and inhibits the overshoot. Thus, the controller exhibits a faster response with less overshoot. In order to robustify the controller against external perturbations, the information regarding the changes in $e_\omega$ and $\dot{e}_\omega$ is incorporated in the SiF, given by Eq. (4), to tune the PI coefficients [12].

$$K_z\left(e_\omega,\dot{e}_\omega\right)=k_{z1}+\frac{k_{z2}-k_{z1}}{1+\ exp\left(\alpha_z.e_\omega.sgn\left(\dot{e}_\omega\right)\right)},\ z\ is\ 'p'\ or\ 'i', \tag{4}$$

where sgn($\dot{e}_\omega$) is the signum function that extracts the sign of $\dot{e}_\omega$, $\alpha_z$ is the variation rate of proportional or integral nonlinear gain functions, and $k_{z1}$ and $k_{z2}$ are the minimum and maximum values of proportional and integral gain, respectively. These parameters are experimentally tuned via trial-and-error method and their fitness is evaluated by the function given in Eq. (5).

$$Fitness=\frac{1}{1+OS+T_{rise}+T_{settling}+|E_{ss}|}, \tag{5}$$

where OS is the overshoot of the response, $T_{rise}$ is the rise time, $T_{settling}$ is the settling time (within $\pm 1\%$ of final value), and $E_{ss}$ is the root-mean-square (RMS) value of the steady state $e_\omega$. The parametric estimates and their fitness values are recorded in Table 1. The waveform of the SiF (for different values of variation rate) is shown in Figure 3.

**Table 1.** Parameter identification of the sigmoidal function.

| Controller | | $k_{p1}$ | $k_{p2}$ | $\alpha_p$ | $k_{i1}$ | $k_{i2}$ | $\alpha_i$ |
|---|---|---|---|---|---|---|---|
| NPI | Parameter value | 0.055 | 0.175 | 2.23 | 0.652 | 1.95 | 1.94 |
| | Fitness value | 0.429 | 0.638 | 0.613 | 0.527 | 0.419 | 0.664 |
| ANPI | Parameter value | 0.055 | 0.175 | Auto-tuned | 0.652 | 1.95 | Auto-tuned |
| | Fitness value | 0.429 | 0.638 | - | 0.527 | 0.419 | - |

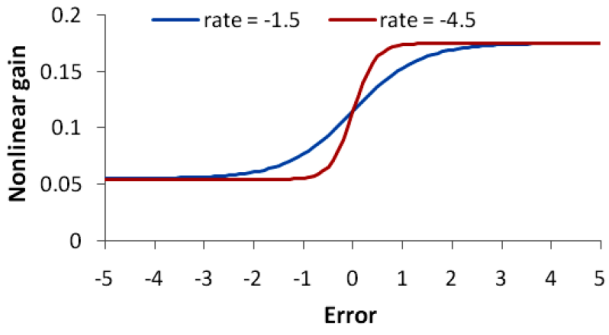


**Figure 3.** Wave form of sigmoidal function.

## 5. EKF-based self-regulation mechanism

The Kalman filter is a stochastic estimator. It approximates the system states and predicts the measurement that could result from that approximation. The process occurs recursively in real time until the algorithm

has converged to deliver the best estimate. When it is linearized about the current mean and covariance, it is denoted as the EKF [24]. The variation rates ($\alpha_z$) of the proportional and integral nonlinear functions are dynamically optimized via EKF to minimize the error. The nonlinear function tunes the PI coefficients after receiving the estimate of $\alpha_z$. Consider a nonlinear system represented by Eqs. (6) and (7).

$$x_{n+1}=f(x_n)+g_n \tag{6}$$

$$d_n= h\,(x_n)+v_n \tag{7}$$

where $x_n$ is the system state vector at the nth instance, $g_n$ is the process noise, $d_n$ is the observation vector, $v_n$ is the observation noise, and h(.) is the nonlinear function of the state. Since the objective is to update the variation rates of the proportional and integral functions, the state vector of the system is represented by Eq. (8).

$$x_n=\left[\begin{array}{cc} \alpha_p & \alpha_i \end{array}\right] \tag{8}$$

The jacobians of the state-transition matrix, $F_n$, and the observation matrix, $H_n$ (measurement model with respect to the estimated states), of the filter are represented via Eqs. (9) and (10), respectively.

$$F_n= \tag{9}$$

$$H_n=\frac{\partial h(x_n)}{\partial x_n}=\frac{\partial \omega}{\partial x_n}=\left[\begin{array}{cc} \frac{\partial \omega}{\partial \alpha_p} & \frac{\partial \omega}{\partial \alpha_i} \end{array}\right] \tag{10}$$

The updated covariance estimate of the prediction error, $P_{n+1}$, is given by Eq. (11). The near optimal Kalman gain, $K_n$, is given by Eq. (12). The updated state estimate of the system, $\hat{x}_{n+1}$, is given by Eq. (13).

$$P_{n+1}= P_n-K_nH_nP_n+Q_n \tag{11}$$

$$K_n=P_nH_n^T\left(R_n+H_nP_nH_n^T\right)^{-1} \tag{12}$$

$$\hat{x}_{n+1}=\hat{x}_{n-1}+K_n\left(d_{n-1}-H_n\hat{x}_{n-1}\right) \tag{13}$$

where $\hat{x}_n$ is the state estimate at the nth instance, $P_n$ is the covariance of the prediction error, and $Q_n$ and $R_n$ are the covariance matrices associated with $g_n$ and $v_n$, respectively. The partial derivative of the system output with respect to $\alpha$ is calculated via Eq. (14).

$$\frac{\partial \omega_r}{\partial \alpha_Z}=\frac{\partial \omega_r}{\partial V_m}\times\frac{\partial V_m}{\partial K_Z}\times\frac{\partial K_Z}{\partial \alpha_Z} \tag{14}$$

The relationship of the partial derivative expressions given in Eq. (14) is elaborated in Eqs. (15)–(17), respectively.

$$\frac{\partial \omega}{\partial V_m}=\frac{\omega\,(t)-\omega(t-1)}{V_m\,(t)-V(t-1)} \tag{15}$$

$$\frac{\partial V_m}{\partial K_Z}=\left\{\begin{array}{ll} \Delta k_p\times e\,(t)\,, & proprotional-tuner \\ \Delta k_i\times\int e\,(t)\;dt, & integral-tuner \end{array}\right. \tag{16}$$

$$\frac{\partial K_Z}{\partial \alpha_z}=-\frac{k_{z2}.e_\omega.sgn\left(\dot{e}_\omega\right).exp\left(\alpha_z.e_\omega.sgn\left(\dot{e}_\omega\right)\right)}{\left(1+\exp\left(\alpha_z.e_\omega.sgn\left(\dot{e}_\omega\right)\right)\right)^2} \tag{17}$$

## 6. Tests and results

The hardware-in-the-loop feature of the experimental setup serves to acquire the analogue measurements of $\omega_r$ and apply the voltage control command ($V_m$) to the motor [25]. The graphical interface of the LABVIEW application is used to apply the reference signals and record the corresponding response of the motor. The speed control performance of the ANPI controller is analyzed in real time against an NPI controller (having fixed variation rates) and a linear PI controller via five different test cases. These test cases and their experimental results are presented as follows.

**Test case A:** A step-input of 100 rad/s is applied to the motor and the consequent responses of the controllers are shown in Figures 4, 5, and 6. The variations in $\alpha_p$ and $\alpha_i$, due to changes in error dynamics, are illustrated in Figure 7.

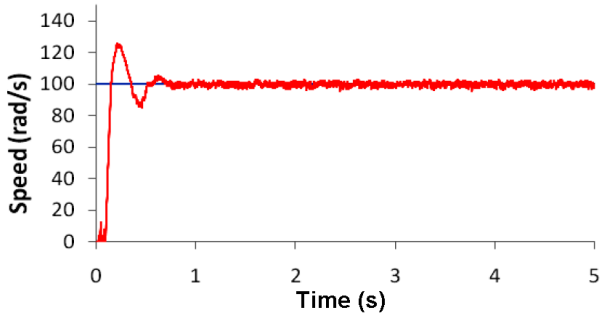


**Figure 4.** Step response of PI without disturbance.

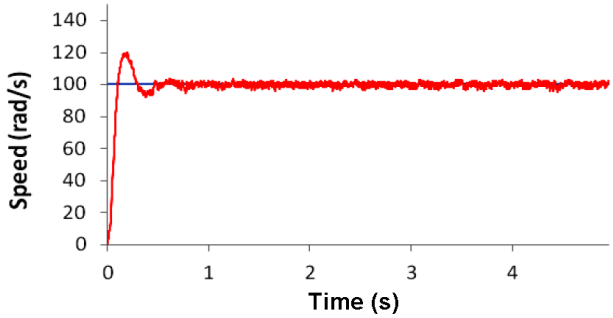


**Figure 5.** Step response of NPI without disturbance.

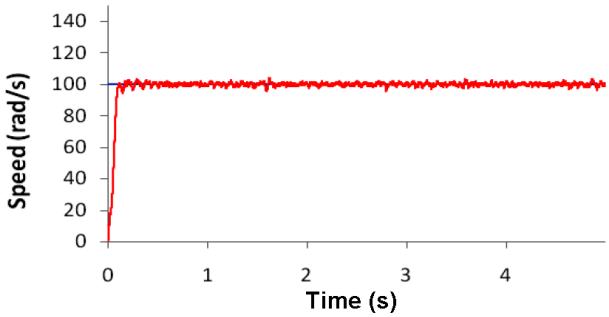


**Figure 6.** Step response of ANPI without disturbance.

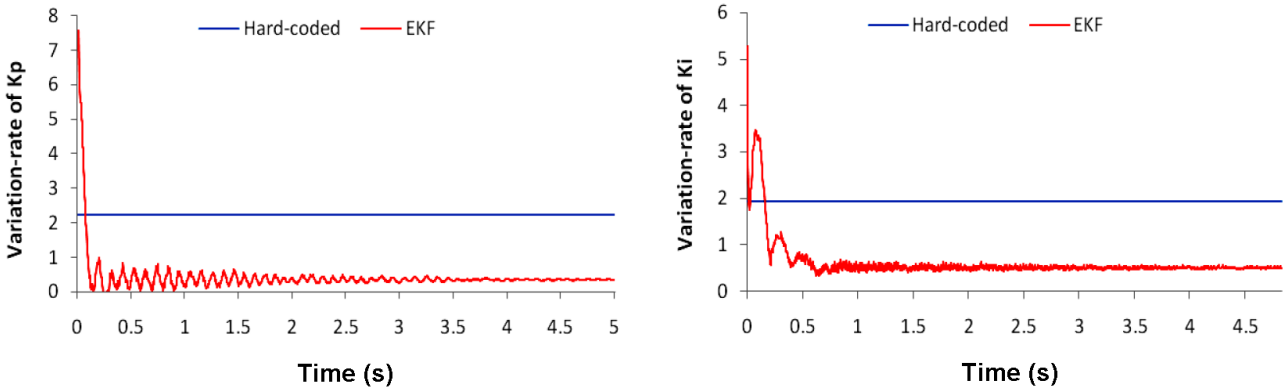


**Figure 7.** Self-regulation of variation rate of $K_p$ and $K_i$ without disturbance.

**Test case B:** Bounded impulsive disturbance signals, having amplitudes of $\pm 20$ rad/s, are randomly applied to perturb the motor's step response. The applied disturbance signals (blue) and the resulting variations (or sudden peaks) occurring in the responses (red) are illustrated in Figures 8, 9, and 10. The changes in $\alpha_p$ and $\alpha_i$, introduced by EKF to mitigate these disturbances, are shown in Figure 11.

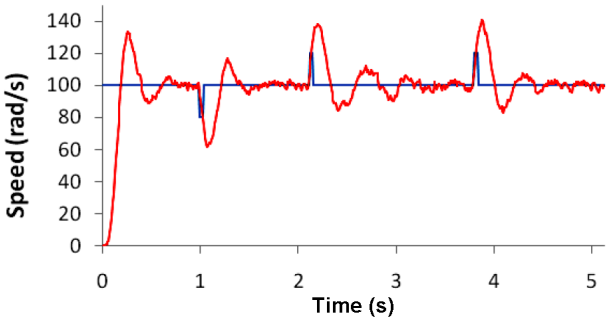


**Figure 8.** Step response of PI with disturbance.

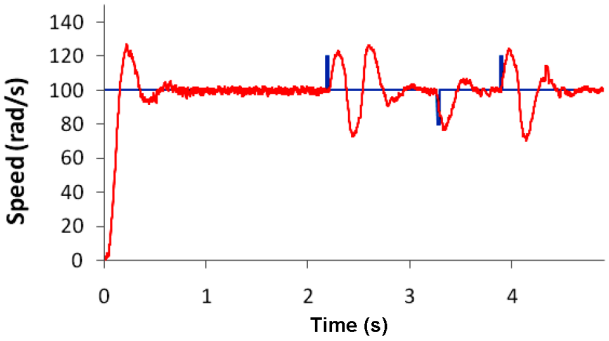


**Figure 9.** Step response of NPI with disturbance.

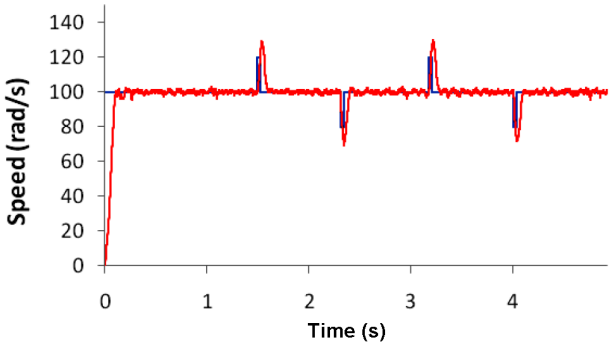


**Figure 10.** Step response of ANPI with disturbance.

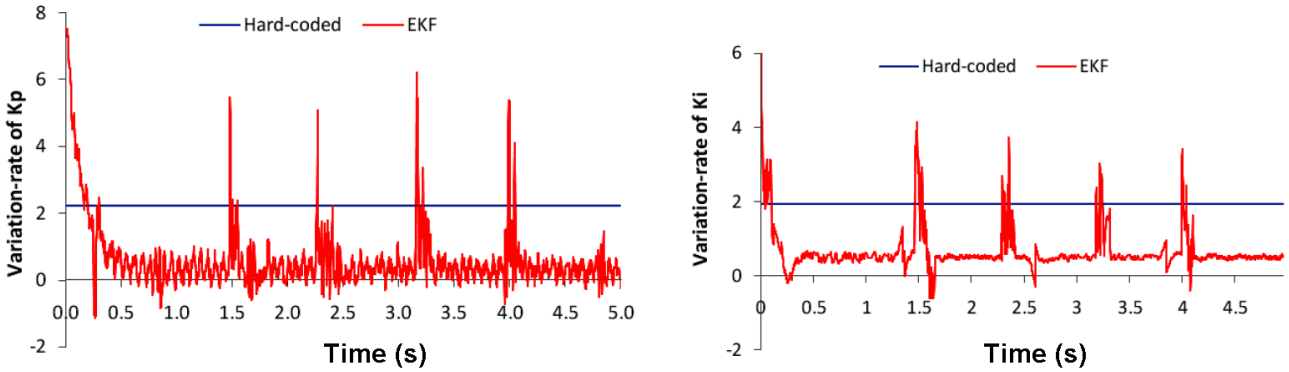


**Figure 11.** Self-regulation of variation rate of $K_p$ and $K_i$ with disturbance.

**Test case C:** The controllers are tested under varying load torque by applying a step input of 100 rad/s to the motor. The load torque is incremented, in steps, by manually applying a braking force to the disc spinning on the motor shaft. As the load torque increases, the controller proportionally increases the $V_m$ in order to regulate the angular speed at 100 rad/s. A specimen of the variation in $V_m$ due to the application

of incremental load torque is shown in Figure 12. A similar scenario is used to test each controller. The corresponding controller responses are shown in Figures 13, 14, and 15.

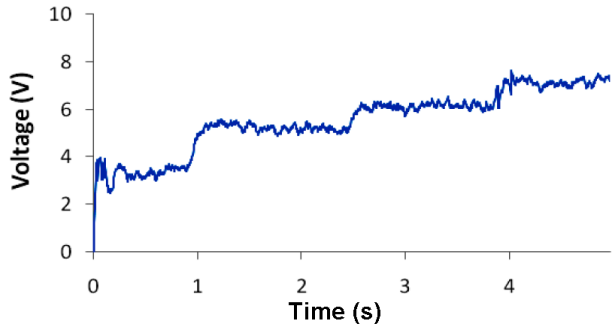


**Figure 12.** Variation in $V_m$ due to load torque application.

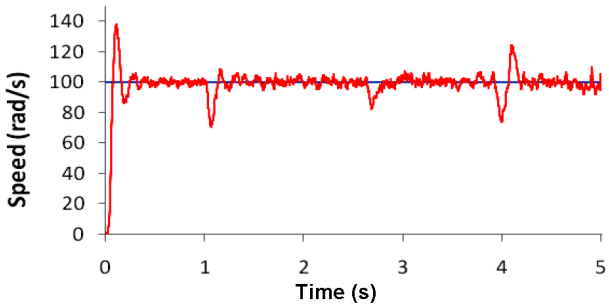


**Figure 13.** Speed regulation with PI controller under varying load torque.

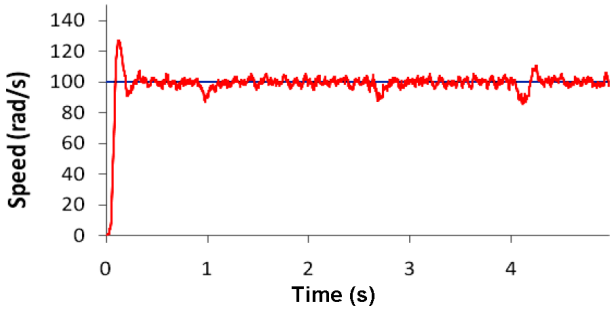


**Figure 14.** Speed regulation with NPI controller under varying load torque.

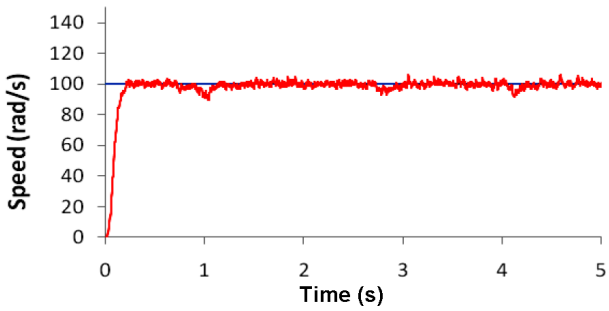


**Figure 15.** Speed regulation with ANPI controller under varying load torque.

**Test case D:** A square-wave signal swinging between 100 rad/s and 20 rad/s at a frequency of 0.25 Hz is applied to the motor. The responses are shown in Figures 16, 17, and 18. The disturbance signals (as used in Test B) are also applied to this system. The resulting peaks in the system's response (red) are illustrated in Figures 19, 20, and 21.

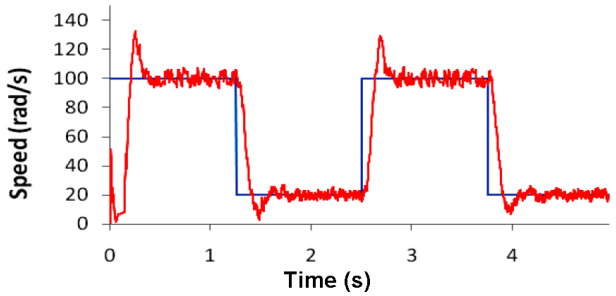


**Figure 16.** Square-wave response of PI without disturbance.

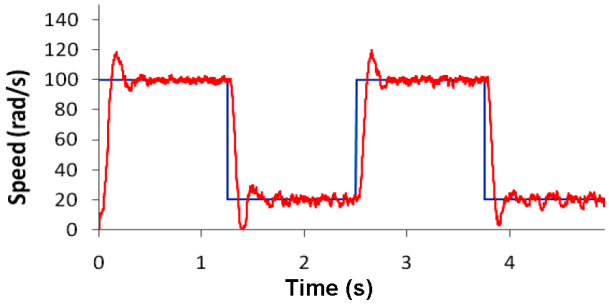


**Figure 17.** Square-wave response of NPI without disturbance.

**Test case E:** A triangular-wave signal varying between 100 rad/s and 20 rad/s at a frequency of 0.25 Hz is applied to the motor. The responses are shown in Figures 22, 23, and 24. The disturbance signals (as used in Test B) are also applied to this system. The resulting peaks in the system's response (red) are illustrated in Figures 25, 26, and 27.

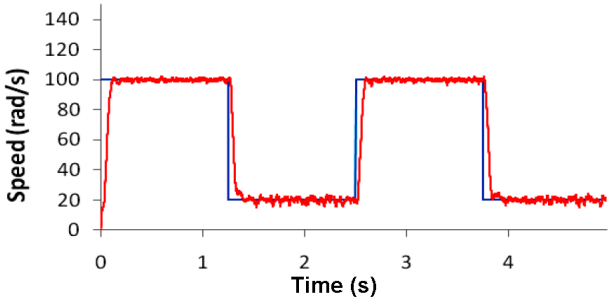


**Figure 18.** Square-wave response of ANPI without disturbance.

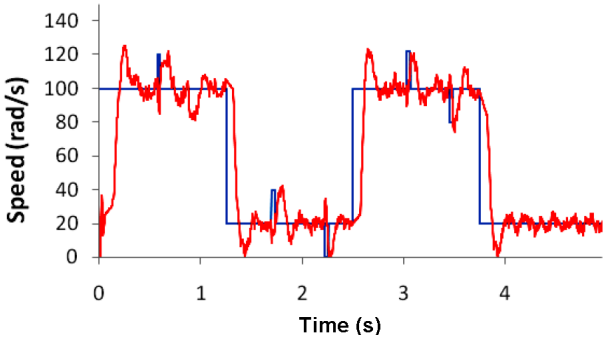


**Figure 19.** Square-wave response of PI with disturbance.

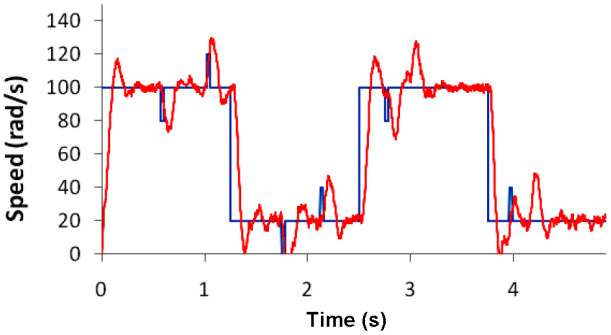


**Figure 20.** Square-wave response of NPI with disturbance.

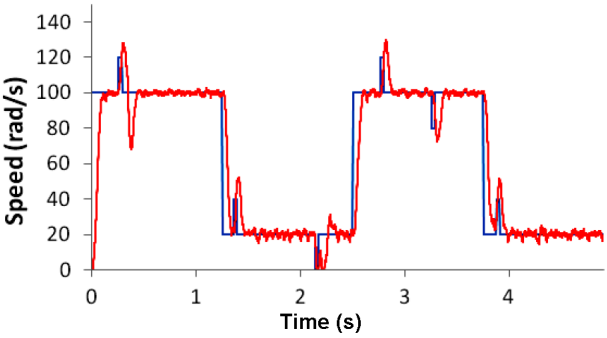


**Figure 21.** Square-wave response of ANPI with disturbance.

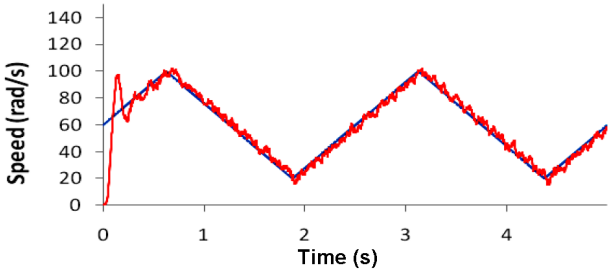


**Figure 22.** Triangular-wave response of PI without disturbance.

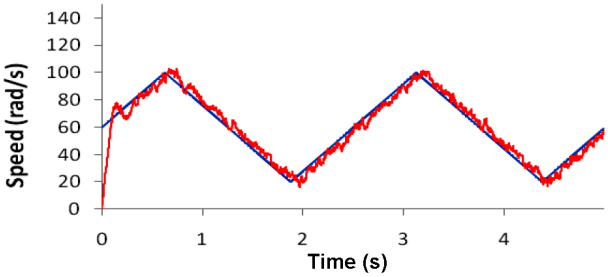


**Figure 23.** Triangular-wave response of NPI without disturbance.

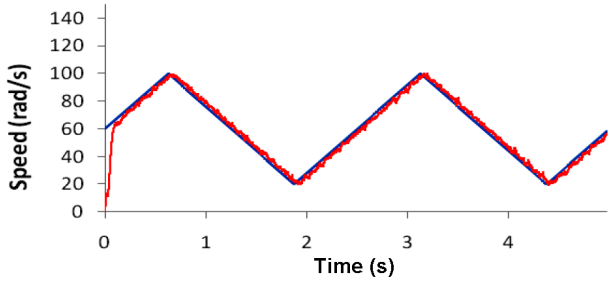


**Figure 24.** Triangular-wave response of ANPI without disturbance.

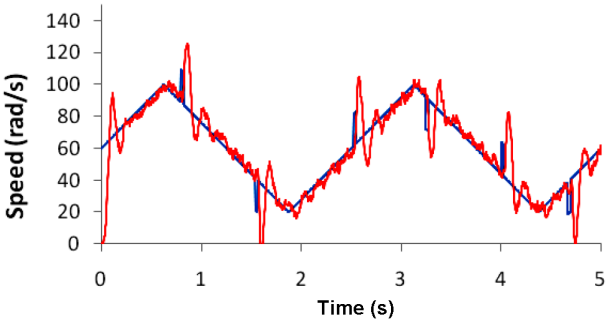


**Figure 25.** Triangular-wave response of PI with disturbance.

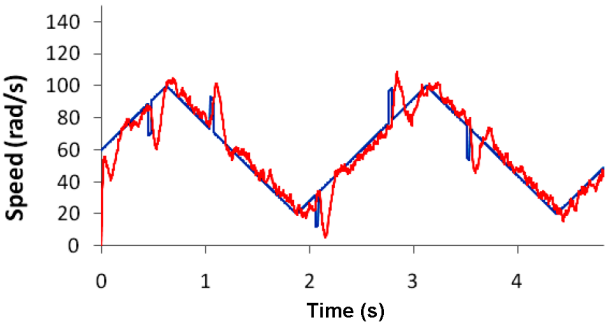


**Figure 26.** Triangular-wave response of NPI with disturbance.

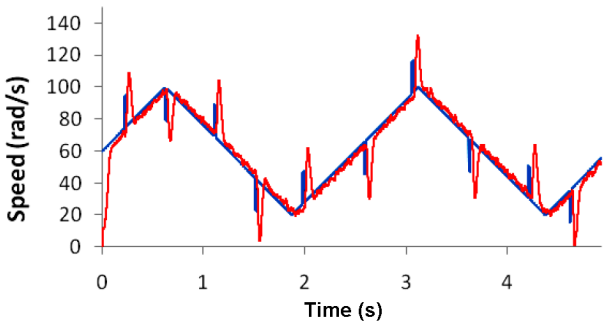


**Figure 27.** Triangular-wave response of ANPI with disturbance.

The comparison of the response graphs attained for the three controllers (via tests A, B, and C) is given in Table 2. The average time taken by the motor to return to its reference speed, after a disturbance (or load torque) is applied, is defined as $T_{recovery}$. The MAE is the maximum-absolute-error incurred in the response upon the application of a disturbance (or load torque).

**Table 2.** Comparative performance assessment of controllers.

| Controller | Test A | | | | Test B | | Test C | |
|---|---|---|---|---|---|---|---|---|
| | $T_{rise}$ (s) | OS (%) | $T_{settling}$ (s) | $E_{ss}$ (rad/s) | MAE (rad/s) | $T_{recovery}$ (s) | MAE (rad/s) | $T_{recovery}$ (s) |
| PI | 0.29 | 29.18 | 0.80 | 2.21 | 37.05 | 0.48 | 27.62 | 0.32 |
| NPI | 0.24 | 20.26 | 0.72 | 2.13 | 22.15 | 0.29 | 15.54 | 0.23 |
| ANPI | 0.13 | 3.93 | 0.21 | 1.10 | 27.31 | 0.11 | 9.86 | 0.18 |

## 7. Conclusions

This paper addresses the self-tuning of nonlinear PI gain functions to regulate the speed of a DC motor. The variation rates ($\alpha$) are adaptively tuned via EKF. The test results demonstrate that the ANPI controller yields a superior performance, when compared with a generic NPI and a linear PI controller. It effectively brings the motor to the reference speed quickly and nullifies the overshoot, even in the presence of bounded disturbances, varying load torques, or rapidly changing reference speed.

## References

[1] Shahgholian G, Maghsoodi M, Mahdavian M, Janghorbani M, Azadeh M, Farazpey S. Analysis of speed control in DC motor drive by using fuzzy control based on model reference adaptive control. In: 2016 13th International Conference on Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology; 28 June–1 July 2016; Chiang Mai, Thailand: IEEE. pp. 1-6.

[2] Visioli A. Practical PID Control. London, UK: Springer-Verlag, 2006.

[3] Khan HS, Kadri MB. DC motor speed control by embedded PI controller with hardware-in-loop simulation. In: 2013 3rd International Conference on Computer, Control & Communication; 22–26 September 2013; Karachi, Pakistan: IEEE. pp. 1-4.

[4] Singh D, Singh N, Singh B, Prakash S. Optimal gain tuning of PI current controller with parameter uncertainty in DC motor drive for speed control. In: 2013 Students Conference on Engineering and Systems; 12–14 April 2013; Allahabad, India: IEEE. pp. 1-6.

[5] Kanojiya GR, Meshram PM. Optimal tuning of PI controller for speed control of DC motor drive using particle swarm optimization. In: 2012 International Conference on Advances in Power Conversion and Energy Technologies; 2–4 August 2012; Andhra Pradesh, India: IEEE. pp. 1-6.

[6] Augustine A, Paul E, Prakash RD. Voltage regulation of STATCOM using fuzzy self tuning PI controller. In: 2016 International Conference on Circuit, Power and Computing Technologies; 18–19 March 2016; KK Dist, India: IEEE. pp. 1-7.

[7] Zhou Y, Shang W, Liu M, Li X, Zeng Y. Simulation of PMSM vector control based on a self-tuning fuzzy PI controller. In: 2015 8th International Conference on Biomedical Engineering and Informatics; 14–16 October 2015; Shenyang, China: IEEE. pp. 609-613.

[8] Simhachalam D, Mudi RK. Self-tuning fuzzy PI controller for integrating and non-linear processes. In: 2014 International Conference on Advances in Electrical Engineering; 9–11 January 2014; Tamilnadu Vellore, India: IEEE. pp. 1-4.

[9] Isayed BM, Hawwa MA. A nonlinear PID control scheme for hard disk drive servosystems. In: 2007 15th Mediterranean Conference on Control & Automation; 27–29 July 2007; Athens, Greece: IEEE. pp. 1-6.

[10] Guo B, Hu L, Bai Y. A nonlinear PID controller with tracking differentiator applying in BLDCM servo system. In: 2012 7th International Power Electronics and Motion Control Conference; 2–5 June 2012; Harbin, China: IEEE. pp. 2467-2471.

[11] Shang WW, Cong S, Li ZX, Jiang SL. Augmented nonlinear PD controller for a redundantly actuated parallel manipulator. Adv Robotics 2009; 23: 1725-1742.

[12] Xu Y, Holerback JM, Ma D. A non-linear PD controller for force and contact transient control. IEEE Contr Syst Mag 1995; 15: 15-21.

[13] Simon D. Training fuzzy systems with the extended Kalman filter. Fuzzy Set Syst 2002; 132: 189-199.

[14] Sakalli A, Beke A, Kumbasar T. Gradient descent and extended Kalman filter based self-tuning interval type-2 fuzzy PID controllers. In: International Conference on Fuzzy Systems; 24–29 July 2016; Vancouver, Canada: IEEE. pp. 1592-1598.

[15] Shadkam M, Mojallali H, Bostani Y. Speed control of DC motor using extended kalman filter based fuzzy PID. IJIEE 2013; 3: 109-112.

[16] Ahn KK, Truong DQ. Online tuning fuzzy PID controller using robust extended Kalman filter. J Process Contr 2009; 19: 1011-1023.

[17] Ghosh A, Sen S, Dey C. Rule reduction of a neuro-fuzzy PI controller with real-time implementation on a speed control process. In: 2015 2nd International Conference on Computer and Communication Technologies; 25–27 September 2015; Allahabad, India. pp. 445-458.

[18] Alli K, Ogboi CI, Ale D, Ajibade AO, Oladipo F. A Labview based online control experiments for student's learning. In: 2015 Proceedings of the World Congress on Engineering and Computer Science; 21–23 October 2015; San Francisco, CA, USA: pp. 308-313.

[19] Engle BJ, Watkins JM. A software platform for implementing digital control experiments on the Quanser DC motor control trainer. In: 17th International Conference on Control Applications; 3–5 September 2008; Texas, USA: IEEE. pp. 510-515.

[20] Abbas AN. Design and implementation of close loop DC motor speed control based on Labview. IJERSTE 2014; 3: 354-361.

[21] Munadi M, Akbar MA, Naniwa T, Taniai Y. Model reference adaptive control for DC motor based on Simulink. In: 2016 6th International Annual Engineering Seminar; 1–3 August 2016; Yogyakarta, Indonesia: pp. 101-106.

[22] Seraji H. A new class of non-linear PID controller with robotic applications. J Robotic Syst 1998; 15: 161-181

[23] Pedroso MD, Nascimento CB, Tusset AM, Kaster MS. Performance comparison between nonlinear and linear controllers applied to a buck converter using poles placement design. In: 2013 15th European Conference on Power Electronics and Applications; 2–6 September 2013; Lille, France: pp. 1-10.

[24] Chui CK, Chen G. Kalman Filtering: with Real-Time Applications. 3rd ed. Berlin, Germany: Springer, 2013.

[25] Muruganandhan S, Jayabaskaran G, Bharathi P. Labview-NI ELVIS II based speed control of DC motor. IJETT 2013; 4: pp. 811-814.