JPE 18-3-8

https://doi.org/10.6113/JPE.2018.18.3.723
ISSN(Print): 1598-2092 / ISSN(Online): 2093-4718

Comparative Analysis of Integer-order and Fractional-order Proportional Integral Speed Controllers for Induction Motor Drive Systems

Adil Khurram*, Habibur Rehman†, Shayok Mukhopadhyay*, and Daniyal Ali*

†,*Department of Electrical Engineering, American University of Sharjah, Sharjah, United Arab Emirates

Abstract

Linear proportional–integral (PI) controllers are an attractive choice for controlling the speed of induction machines because of their simplicity and ease of implementation. Fractional-order PI (FO–PI) controllers, however, perform better than PI controllers because of their nonlinear nature and the underlying iso-damping property of fractional-order operators. In this work, an FO–PI controller based on the proposed first-order plus dead-time induction motor model and integer-order (IO) controllers, such as Ziegler–Nichols PI, Cohen–Coon PI, and a PI controller tuned via trial-and-error method, is designed. Simulation and experimental investigation on an indirect field-oriented induction motor drive system proves that the proposed FO–PI controller has better speed tracking, lesser settling time, better disturbance rejection, and lower speed tracking error compared with linear IO–PI controllers. Our experimental study also validates that the FO–PI controller maximizes the torque per ampere output of the induction machine and can effectively control the motor at low speed, in field-weakening regions, and under detuned conditions.

Key words: Detuning, Field weakening, Fractional-order controller, Induction motor, Nonlinear controller, Speed regulator

NOMENCLATURE

V_a, V_b, V_c	Three-phase PWM output line voltage
V_{dc}	Three-phase inverter DC bus
i_a, i_b	Induction motor current in the ABC frame of reference
i_{ds}, i_{qs}	Direct axis and quadrature axis currents
i^*_{ds}, i^*_{qs}	Reference direct axis and quarature axis currents
ω_r	Induction motor speed
ω^*_r	Reference speed
ω^*_{sl}	Induction motor slip frequency
ω^*_e	Induction motor electrical frequency
θ^*_e	Rotor angle
R_S	Stator resistance
R_R	Rotor resistance
L_{LS}	Stator self-inductance
L_{LR}	Rotor self-inductance
L_M	Mutual inductance
T_R	Rotor time constant
L_R	Rotor inductance
T_E	Electromagnetic torque
J	Motor inertia
B	Coulomb friction
P	Number of pole pairs
PI	Proportional–integral controller
E	Error signal provided to controller
FPDT	First-order plus dead-time
K_{PRCS}	FPDT process gain
L	FPDT dead time
T	FPDT time constant
α	Order of fractional operator
K_P	Proportional gain
K_I	Integral gain

Manuscript received Mar. 12, 2017; accepted Jan. 8, 2018
Recommended for publication by Associate Editor Zheng Wang.
†Corresponding Author: rhabib@aus.edu
Tel: +971-50-733-0819, American University of Sharjah
*Dept. of Electrical Eng., American Univ. of Sharjah, United Arab

I. INTRODUCTION

An accurate, rapid, and robust speed controller design is critical for high-performance induction motor drive systems.

The two main types of controllers preferred for torque/speed control of various machines are linear proportional–integral (PI) controllers and nonlinear controllers. Integer-order PI (IO–PI) controllers are an attractive choice for the industry because of their simplicity and ease of implementation. Several variations of existing tuning rules for linear PI controllers have been proposed in literature. The rules provided by Ziegler–Nichols (ZN) [1] are widely accepted and adopted in industrial applications. Cohen–Coon (CC) [2], however, have proposed modifications to these rules. The performance of linear proportional integral derivative (PID) controllers, though widely used by the industry, degrades because of inherent fundamental limitations, such as saturation and reduced stability margin [3], particularly when such controllers are used for nonlinear plants like induction machines. In addition, PID controller performance deteriorates because of the variations in motor parameters. Induction motors and permanent-magnet synchronous motors are nonlinear plants and are expected to perform better with nonlinear controllers than with linear controllers. Therefore, nonlinear controllers are being investigated [3]-[19] for these types of machines.

Recently, auto-disturbance rejection controller (ADRC) has been explored because of its capability to estimate and compensate for external disturbance without exact knowledge of the plant parameters [3]-[5]. ADRC employs generalized derivatives and generalized function-based extended state observers (ESOs) to estimate disturbance [4]. However, the incorrect selection of ADRC parameters can lead to a divergence of the disturbance estimator. The use of second- and third-order ADRCs with third-order dynamic equations for ESOs also leads to high computational cost [5]. Passivity-[7] and Hamiltonian-based [8] nonlinear controllers are investigated for improving the performance of induction motor drive systems. These controllers are also dependent on the system parameters [4]. Back-stepping, adaptive back-stepping, and ESOs have been proposed to compensate for the variations in controller parameters [4].

Nonlinear model predictive control (NMPC) schemes [8]-[11] adjust the control input based on the time evolution of the system model. NMPC is dependent on the system model because it evaluates beforehand the control signal for tracking a known future reference. However, NMPC schemes are computationally intensive because model predictive control requires the solution of an optimal control problem at each sampling instant [12].

The robustness under parameter variation and fast dynamic response of a sliding-mode controller [13] make it an attractive choice for electric drives applications [13]-[16]. However, when used for speed regulation, a sliding-mode controller introduces chattering [15], [16], which causes torque ripples in an AC machine. Adaptive fuzzy sliding-mode controllers are used to avoid the chattering problem [16]. Compared with PI controllers, fuzzy logic controllers [17], [19] provide a systematic way to incorporate the experience of the control engineer into the controller, making it more robust and leading to better performance even under certain parameter variations. However, such approaches are dependent upon the skills of the engineer in estimating the disturbance effect on the plant output.

The motor drive system's performance in the field-weakening region plays a vital role in high-speed applications, such as traction motor drives. In such applications, the drive system is operated below the base speed only during starting and low-speed operation [20]. Therefore, several field-weakening schemes [21] have been proposed in literature, including $1/\omega_m$, model-based methods, and voltage detection methods [20]. The $1/\omega_m$ method [22] reduces the flux proportionately as the speed increases beyond the base speed [21]. The model-based scheme [22] uses the induction motor model to determine the flux setting of the drive system. The voltage detection method [23] measures the voltage and considers the maximum operating limits of the inverter while calculating the optimum flux reference.

Nonlinear ADRCs [3]-[5] are sensitive to the controller parameters. NMPC [8]-[11] is computationally intensive. Although sliding mode [13]-[16] is a robust controller, it suffers from the chattering problem, which induces torque ripples. Fuzzy logic controllers [17]-[19] perform better than IO–PI controllers but requires rigorous tuning and are computationally intensive. Therefore, the present work proposes a nonlinear fractional-order PI (FO–PI) controller for controlling the speed of an induction motor drive system. The proposed controller can perform better than a linear PI controller, is simpler to implement, and is less dependent on motor parameters compared with other nonlinear controllers [3]-[19]. The major advantage of the FO–PI controller is it can compensate for plant nonlinearities and changes in parameters and has good disturbance rejection characteristics, even though a first-order model is used for a plant (i.e., an induction motor in this study). A minor drawback is that the implementation of such controllers may appear difficult; however, readily available packages [24] make the job of implementing such controllers relatively easy. Our preliminary results showing the potential of FO–PI controller are shared in [25], [26]. The current paper presents a detailed performance analysis of FO–PI controllers and a comparison with PI controllers. The four main contributions of this work are as follows. (i) As per the authors' knowledge, this work is the first to investigate the full potential of a nonlinear FO–PI controller for an induction motor drive system through simulation and experimental study and provide a detailed comparison of FO–PI and conventional IO–PI controllers. (ii) A simple procedure for gain tuning of the FO–PI controller based on the tuning rules provided in [27] is designed for the regulation of induction motor speed. (iii) This study shows

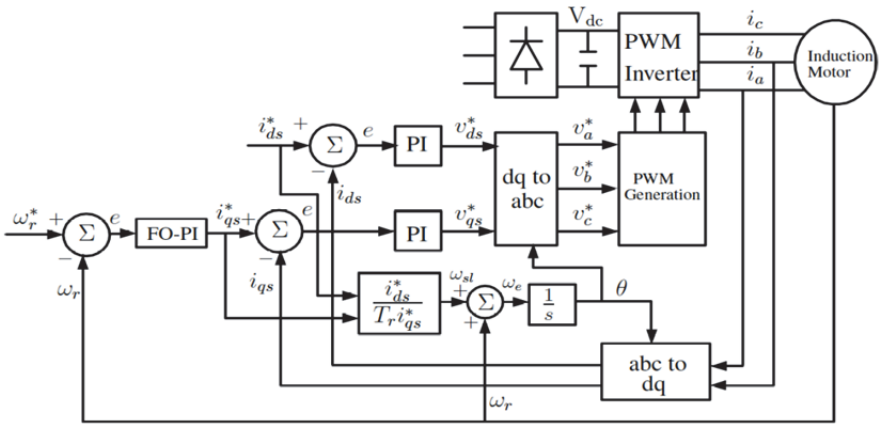

Fig. 1. Fractional-order Proportional–Integral-based IFO drive system.

for the first time that an FO–PI controller can control a detuned motor's speed very well and can maximize the torque per ampere output of the induction machine. Furthermore, the performance of the proposed control scheme is evaluated in the field-weakening region of the induction motor drive system. (iv) The above work is performed based on the FO–PI controller gain tuning using a simple first-order model, which shows that the proposed controller works effectively for an induction motor even if nonlinear effects like magnetic saturation and change of motor parameters over time are not explicitly considered in the model.

The remainder of the paper is organized as follows. The details on fractional-order (FO) and integer-order (IO) controllers, as well as the tuning rules for designing these controllers, are provided in Section II. The simulation and experimental results comparing the performance of different PI controllers with that of the FO–PI controller are presented in Sections III and IV, respectively. Finally, the concluding remarks and the contributions made by this work are shared in Section V.

II. Design of Integer-order and Fractional-order Controllers

An indirect field-oriented (IFO) induction motor drive system is shown in Fig. 1. The proposed IFO drive system in this work is controlled by using either an IO–PI or an FO–PI controller. The design of IO controllers, such as ZN–PI controller, CC–PI controller, and FO–PI controller, is carried out in two steps. The first step approximates the induction motor with a first-order plus dead-time (FPDT) model. This model is then used to obtain the controller parameters of IO and FO controllers. The procedure for obtaining the FPDT model of the induction motor, the IO controller, the proposed FO controller, and the tuning rules for the FO–PI controller are detailed in the following subsections A, B, C, and D, respectively.

A. FPDT Model of an IFO Drive System

The electromagnetic torque (T_e) for an IFO drive system and an induction motor mechanical model are represented by Eqs. (1) and (2), respectively.

$$T_e = \frac{3pL_m^2}{2L_r} i_{ds} i_{qs} \tag{1}$$

$$J\frac{d\omega_r}{dt} = T_e - B\omega_r \tag{2}$$

In the above equations, L_m is the mutual inductance, L_r is the rotor inductance, i_{ds} is the flux current, p is the number of pole pairs, J is the motor inertia, and B is the Coulomb friction. Eqs. (1) and (2) are used to derive the open-loop transfer functions related to motor output speed (ω_r) and torque current (i_{qs}). This transfer function [$G_p(s)$] is given by Eq. (3). The approximate transfer function is obtained by applying a step command current i_{qs}. An inner PI loop ensures that the actual current applied to the motor i_{qs} equals the commanded current. The step speed response produced by this torque command current is shown in Fig. 2. This step response is used to approximate the transfer function [28] with an FPDT model, as given by Eq. (4).

$$G_p(s) = \frac{\omega_r}{i_{qs}} = \frac{\frac{3pL_m^2}{2Lr} i_{ds}}{Js+B} \tag{3}$$

$$G_{approx}(s) = \frac{Kprcs}{Ts+1} e^{-Ls} \tag{4}$$

The transfer function parameters are dead time (L), time constant (T), and process gain (K_{prcs}). The values of these parameters are obtained by feeding the step response shown in Fig. 2 to the PID Tuner toolbox in MATLAB R14a. The values of L, T, and K_{prcs} are found to be 0.03062 seconds, 9.43 seconds, and 609:43 *rad/sec*, respectively, as given by Eq. (5).

$$G_{ident}(s) = \frac{609.43}{9.43s+1} e^{-0.03062} \tag{5}$$

The actual response in Fig. 2 matches the FPDT model response, which shows the accuracy of the approximate first-order induction motor model. The FPDT model when obtained experimentally is more accurate compared with that derived through the simulation because the actual test on the motor considers the nonlinearities, friction, and actual inertia

of the motor. Thus, the FPDT model of the induction motor obtained experimentally in Eq. (5) is used both for simulation and experimental studies in this study. Now we proceed with the design of an IO controller for an induction motor drive system.

B. Integer-order Controller

The conventional IO controller defined by Eq. (6) consists of a proportional gain K_p and an integral gain K_i.

$$u(t) = K_p e(t) + K_i \int_0^t e(\tau)\, d\tau \tag{6}$$

The parameter K_o is calculated using Eq. (7) and is used to design the ZN–PI and CC–PI controllers.

$$K_o = \frac{X_o}{K_{prcs}} \times \frac{T}{L} \tag{7}$$

In this equation, X_o is the current command i^*_{qs} used to obtain the FPDT model. The parameter K_o is used to find K_p and K_i both by ZN–PI and CC–PI controllers. The ZN controller gains K_p and K_i are calculated by Eq. (8), in which the parameters L and T are obtained from Eq. (5) and K_o is calculated by Eq. (7).

$$\begin{aligned} K_p &= 0.9K_o \\ T_i &= 3.3L \\ K_i &= \frac{K_p}{T_i} \end{aligned} \tag{8}$$

The CC tuning rules shown in Eq. (9) are applicable to a broader class of plants.

$$\begin{aligned} R &= \frac{L}{T} \\ K_p &= K_o\left(0.9 + \frac{R}{12}\right) \\ K_i &= L\left(\frac{30+3R}{9+20R}\right) \end{aligned} \tag{9}$$

These rules are used to calculate K_p and K_i for the CC tuned controller, where L is the dead time, T is the time constant, and R is the ratio of L and T as defined in Eq. (9). The gains obtained after the application of the ZN and CC tuning rules are tabulated in Table I. For the procedure to determine gains and the order α of the FO–PI controller, please see the sections C and D below. In Table I, the order α = 1 for the CC–PI, ZN–PI, and TE–PI because these are IO controllers, and a full first-order integral is used.

C. Fractional-order Control: A Brief Review

Fractional calculus, which is at the base of FO control, is almost as old a discipline as IO calculus [29]. FO controllers and models have been successfully used on a wide number of experimental platforms, such as DC motor speed control, heat flow, and Tokamak plasma position control [29], to name a few. In all such experimentation, when compared with IO controllers, the FO controllers can better deal with the plant nonlinearities and effects of noise and disturbance, as well as reduce the required control effort. This outcome can be attributed to the fact that traditional PI and PID controllers

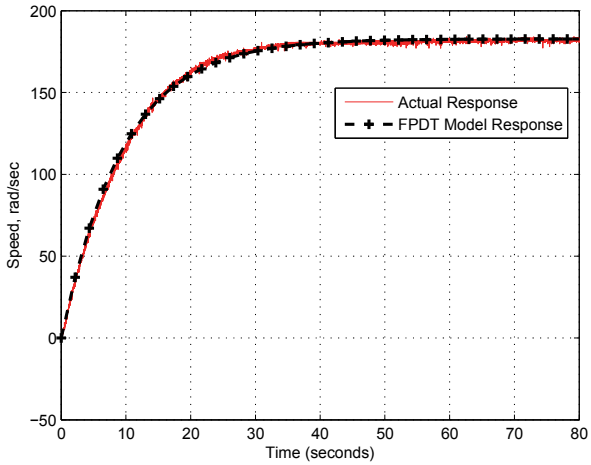

Fig. 2. Step response used to obtain the FPDT model.

have only two or three gains, respectively, which can be tuned. On the other hand, their FO counterparts, namely, FO–PI and FO–PID controllers have three or five parameters, respectively, which can be tuned. The availability of additional parameters that can be tuned [30] makes meeting the following desirable characteristics of a controller easier: (1) specified phase margin, (2) specified gain margin, (3) specified open-loop gain crossover frequency (ω_{cg}), (4) flat phase around ω_{cg}, (5) zero steady-state error, and (6) specific gains at specific frequencies of the four possible closed-loop sensitivity functions.

The flat phase around ω_{cg} is specifically important because in real-life systems, such as an induction motor, motor parameters may change slightly over time because of wear and tear, routine use, and so on. From a control perspective, these changes cause a change in the overall motor transfer function gain. The flat-phase around ω_{cg} is crucial because it guarantees that the phase margin remains unchanged even if the plant gain changes. This effect makes the controller more robust to disturbances and the aging of the plant (motor) under control. Moreover, from basic control theory, introducing a D term in a controller can improve the settling time and even reduce overshoot, though it can also amplify the noise. When an FO–PI controller is used, the implementation [29] of an integral term of order 0.7 may require that a derivative operator of order 0.3 is used. This will have several effects. Some of the benefits caused by the existence of a derivative term are retained, but ill effects owing to disturbances and noise are attenuated compared with a first-order derivative term. These controllers are also saturated slower than conventional PI controllers because an integral term of positive order less than 1 winds up slower than a conventional first-order integrator. The above discussion demonstrates the advantages of using an FO controller over a traditional IO controller. Note that packages such as the "*Ninteger toolbox*" are readily available [24], thus allowing the implementation of FO derivatives/integrals and

associated controllers.

An FO–PI controller given by Eq. (10) is obtained by replacing the IO integral term in the PI controller by an integral term of arbitrary order [31].

$$u(t) = K_p e(t) + K_i({}_aD_t^{\alpha} e(t)) \quad (10)$$

where $u(t)$ is the control output, $e(t)$ is the error between the actual speed and the reference speed, and α is the order of the integral operator. The order α is an additional parameter that can be tuned along with the gains of the FO–PI controller K_p and K_i. The differintegral operator ${}_a^C D_t^{\alpha}$ is used for evaluating the integral of arbitrary order, and the FO Caputo derivative-based implementations with a and t are used as the upper and lower terminals, respectively, as shown in Eq. (11). The letter C on the left-hand side (LHS) of Eq. (11) is used to show that the operator used is of the Caputo form. Contrary to the definition of a regular IO derivative, which is specified at an instant of time, the differintegral operators are required to be specified over a range of time given by the lower and upper time limits, which are defined by a and t, respectively. a represents a previous instant of time, and t represents the current time instant.

$${}_a^C D_t^{\alpha} f(t) = \frac{1}{\Gamma(n-\alpha)} \int_a^t \frac{f^{(n)}(\tau)}{(t-\tau)^{\alpha+1-n}} d\tau \quad (11)$$

$n-1 < \alpha < n$; $n \epsilon Z$; $\Gamma(.)$ represents the standard Gamma function, that is, $\Gamma(x+1) = x\Gamma(x)$ and $\alpha < \Re$. Note that $f^{(n)}(.)$ represents the n[th]-order derivative of the function $f(.)$. As an example, consider $\alpha = 0.5$, that is, α lies between integers $n-1 = 0$ and $n = 1$. If $a = 0.1$ second, we can write the above Caputo differintegral operator for a given function $f(t)$, which is evaluated at time t as

$${}_{0.1}^C D_t^{0.5} f(t) = \frac{1}{\Gamma(0.5)} \int_{0.1}^t \frac{f^{(1)}(\tau)}{(t-\tau)^{0.5}} d\tau \quad (12)$$

where $f^{(1)}(.)$ represents the first-order derivative of $f(.)$. Note that $f^{(0)}(.)$ represents the function $f(.)$ itself.

The differintegral operator can also be defined using the Grunwald–Letnikov (GL) form given in Eq. (13) or the Riemann–Liouville (RL) form given in Eq. (15) [32]. Similar to Eq. (11), the letters a and t on the LHS of Eq. (13) represent the upper and lower terminals of time, respectively. The letters *GL* on the LHS of Eq. (13) shows that the operator used is of the GL form.

$${}_a^{GL} D_t^{\alpha} f(t) = \lim_{\substack{h \to 0 \\ nh = t-a}} h^{-\alpha} \sum_{k=0}^{n} (-1)^k \binom{\alpha}{k} f(t-kh) \quad (13)$$

where $\binom{\alpha}{k}$ represents binomial coefficients. The n in Eq. (13) is an integer that represents the number of terms to be considered in the summation in Eq. (13); it has nothing to do with n in Eq. (11). The idea for using Eq. (13) is to make the step size h as close to 0 as possible, which can be achieved by increasing the number of terms n in the summation in Eq. (13). As an example, consider $\alpha = 0.5$. If $a = 0.1$ second and $n = 5$, then we can write the GL differintegral operator for a given function f(t), which is evaluated at time t as follows:

$${}_{0.1}^{GL} D_t^{0.5} f(t) = \left(\frac{t-0.1}{5}\right)^{-0.5} \sum_{k=0}^{5} (-1)^k \binom{0.5}{k} f(t-kh) \quad (14)$$

where $h = \frac{t-0.1}{5}$ and $\binom{0.5}{k} = \frac{\Gamma(0.5+1)}{\Gamma(k+1)\Gamma(0.5-k+1)}$. Note that $\Gamma(1.5) = 0.8862$ and several other equivalent definitions for calculating binomial coefficients for real numbers are available. However, the one shown above is a general form which allows calculation of binomial coefficients for non-integer values.

Another way to represent the differintegral operator is through the RL form shown in Eq. (15). The letters a and t on the LHS of Eq. (15) represent the upper and lower terminals of time, respectively. The letters *RL* on the LHS of Eq. (15) show that the operator used is of the RL form.

$${}_a^{RL} D_t^{\alpha} f(t) = \left(\frac{1}{\Gamma(m+1-\alpha)}\right) \left(\frac{d}{dt}\right)^{m+1} \times \left(\int_a^t (t-\tau)^{(m-\alpha)} f(\tau)\, d\tau\right) \quad (15)$$

where $\alpha \epsilon \Re$, $m \epsilon Z$, $m \leq \alpha < m+1$, and $m = [\alpha]$, The $[.]$ operator represents the integer part of the real number α. As an example, consider $\alpha = 0.5$. If $a = 0.1$ second, and realizing that in this case $m = [0.5] = 0$, $m+1 = 1$, then we can write the RL differintegral operator for a given function $f(t)$, which is evaluated at time 't' as

$${}_{0.1}^{RL} D_t^{0.5} f(t) = \left(\frac{1}{\Gamma(0+1-0.5)}\right) \left(\frac{d}{dt}\right)^{0+1} \times \left(\int_{0.1}^t (t-\tau)^{(0-0.5)} f(\tau)\, d\tau\right) \quad (16)$$

Note that the operators ${}_a^C D_t^{\alpha}$, ${}_a^{GL} D_t^{\alpha}$, and ${}_a^{RL} D_t^{\alpha}$ are called the differintegral operators because α can take positive or negative real values. As a result, they can act as an FO integral or an FO differentiator. The advantages and disadvantages of each type of operator can be found in any standard reference about fractional calculus [32]. For example, one common disadvantage related to some forms of differintegral operators is that the limit value of the differintegral operator must be determined at the initial time instant or at the time instant equal to the lower terminal value of time. As shown in Eq. (17) the initial condition for the differintegral operator in RL form below, the value b_1 must be found. The value of b_1 will be different for different functions, different orders α, and different lower terminal values a.

$$\lim_{t \to a} {}_a^{RL} D_t^{\alpha} f(t) = b_1 \quad (17)$$

The Caputo form does not have such issues related to initial conditions [29].

D. Tuning Rules for FO–PI Controller

The tuning rules used for the design of an optimum FO–PI controller developed in [27] are a generalized form of maximum sensitivity (M_s) constrained integral gain optimization (MIGO)-based tuning method. They have been modified for FO–PI controllers and are called fractional

MIGO (F-MIGO) tuning rules [27]. The dead time is represented by T in the tuning rules given in Eqs. (18)-(21).

$$\tau = \frac{L}{T+L}, \alpha = \begin{Bmatrix} 1.1 & \tau \geq 0.6 \\ 1.0 & 0.4 \leq \tau \leq 0.6 \\ 0.9 & 0.1 \leq \tau \leq 0.4 \\ 0.7 & \tau < 0.1 \end{Bmatrix} \quad (18)$$

$$K_p = \frac{1}{K_{prcs}}\left(\frac{0.2978}{L+0.000307}\right) \quad (19)$$

$$T_i^* = T\left(\frac{0.8578}{L^2-3.402L+2.405}\right) \quad (20)$$

$$K_i = \frac{K_p}{T_i^*} \quad (21)$$

These rules are developed by imposing a constraint on the maximum load disturbance to output sensitivity (M_s) and the resonance peak of the closed-loop system (M_p). In the Nyquist plot, M_s is represented as a circle of radius $1/M_s$ centered at −1, while M_p is represented as a circle of radius $[M_p/(M_p^2-1)]$ centered at $[-M_p/(M_p^2-1)]$. The integral gain (K_i) of the controller is maximized subject to the constraint that the Nyquist plot of the loop transfer function should encompass both M_s and M_p circles [27]. The FO α depends on the relative dead time τ, as shown in Eq. (18). The authors in [27] conducted several tests on systems with different τ. If the systems have values of τ in different ranges, then different values of α must be chosen according to Eq. (18). For example, if τ is between 0.4 and 0.6 in value, then α must be chosen as 1; otherwise, if τ is between 0.1 and 0.4, then α must be chosen as 0.9, and so on. In our work, the value of τ is determined using the numbers obtained for T and L as shown in Eq. (5). For each type of system with the value of τ in the ranges given in Eq. (18), the authors in [27] conducted tests for various values of α. The best FO (α) for each system is chosen to be the one that gives the minimum integral squared error (ISE). A choice of α works over a certain range of values of the relative dead time τ; these are shown in the tuning rules in Eq. (18). Equations (19)–(21) are used to calculate the gains of an FO–PI controller; these gains are tabulated in Table I. Table I shows that the gains of the FO–PI controller are smaller compared with those of the ZN–PI and CC–PI controllers. The higher gains of the ZN–PI and CC–PI controllers can potentially saturate the controller. Once the gains of a controller (FO–PI, CC–PI, ZN–PI, or TE–PI) are tuned, their values are kept fixed throughout all simulation and experimental tests for fair comparison. If a conventional PI controller is manually tuned, then tuning it for a type of load, rejecting a disturbance, or handling a nonlinearity is possible. However, the tuning rules, namely, ZN–PI, CC–PI, and the FO–PI used in this paper, are well established tuning rules from literature and offer the convenience of tuning once for a variety of circumstances. This study shows that the FO–PI controller operates with lesser effort compared with other PI controllers even though it is tuned only once. In addition, even the conventionally/experimentally tuned (TE–PI) controllers require more control effort compared with FO–PI controllers. Experimental tuning likewise requires operator experience and considerable time investment for tuning. Such time investment for tuning is not required when the FO–PI tuning rules are used.

TABLE I
PI CONTROLLER GAINS

Parameters	FO–PI	CC–PI	ZN–PI	TE–PI
α	0.7	1	1	1
K_p	0.1406	0.4649	0.4649	0.01
K_i	0.0407	4.5993	4.5853	0.02

K_p = Proportional gain, K_i = Integral gain

III. SIMULATION RESULTS

The computer simulations for an IFO drive system (Fig. 1) with a sine triangle pulse-width modulated inverter are developed in the SIMULINK environment. A 175 W prototype induction motor, whose parameters are given in Table II, is used in both simulation and experimental studies. The controller is designed using the FPDT model of the system developed in Section II-A. The resulting gains K_p and K_i and the order are shown in Table I. The order of the integral obtained for the FO–PI controller is 0.7. The integral gains obtained in the cases of the ZN–PI and CC–PI controllers are higher compared with the corresponding integral gain in the case of the FO–PI controller. The TE–PI controller is also used for comparison with other controllers. The K_p and K_i gains obtained using the trial-and-error technique at the rated speed of the test induction motor are also included in Table I.

The controller performance is tested for 1400 RPM square-wave tracking. The simulation results using FO–PI and IO–PI (ZN–PI, CC–PI, and TE–PI) controllers are shown in Figs. 3 and 4. The controller parameters are kept as tabulated in Table I for all the simulations and experimental study. The speed response of all four controllers presented in Figs. 3(a) to (d) show that the FO–PI controller has an overshoot of 8.3%, which is less than that of the three IO controllers. The overshoot is highest in the case of the CC–PI and ZN–PI controllers at 44.25%. The TE–PI controller, on the other hand, has an overshoot of 30.83%, which is lower than those of the CC–PI and ZN–PI controllers but higher than that of FO–PI controller, as shown in Fig. 3(d).

Figs. 4(b) and (c) show that compared with the FO–PI controller, the CC–PI and ZN–PI controllers require the maximum allowable quadrature current (i_{qs}) for a longer period during the transient to track the same reference. The i_{qs} plotted for the TE–PI controller [Fig. 4(d)] shows that it requires less control effort than the CC–PI and ZN–PI controllers, but its performance is poorer compared with the FO–PI controller. The magnitude of i_{qs} in the case of both CC–PI and ZN–PI controllers is 1 A for approximately 3.33

seconds, every time the reference has a step change of 2800 RPM. The FO–PI controller's i_{qs} is 1 A for approximately 1.3 seconds. These findings indicate that the FO–PI controller designed using the tuning rules presented in Section II-D result in lower overshoot and require lesser control effort compared with IO controllers (CC–PI, ZN–PI, and TE–PI).

IV. Experimental Results

The motor drive system setup in the laboratory is shown in Fig. 5, and its specifications are given in Table III. This drive system is used to compare the performance of IO–PI and FO–PI controllers. The controller gains are kept the same as given in Table I. The induction motor drive system is affected by the nonlinearities introduced because of the PWM-based three-phase inverter and the nonlinear nature of the motor itself. Three performance parameters, ISE, IAE, and ITAE as defined by Eq. (22), are used to compare these controllers' capability to accurately track the reference speed. Moreover, the control effort required by the controller for tracking the reference speed is used to evaluate the torque per ampere output of the machine.

A. Evaluation of the Controller's Speed Tracking Performance

The speed tracking performance of the drive system is tested by setting up a square-wave reference of 1400 RPM. The IAE, ISE, and ITAE for the first 20 seconds are plotted in Fig. 6. The IAE, ISE, and ITAE at the end of 20 seconds are 38.75, 2,494, and 1,236, respectively, when the motor speed is regulated by the FO–PI controller. The performance of the CC–PI and ZN–PI controllers is poorer compared with that of the FO–PI controller, with IAE, ISE, and ITAE of 67, 4,574, and 2,114, respectively, for both controllers. These results show that the FO–PI controller performs better than the IO controllers. The poor performance of IO controllers is caused by the high K_p and K_i gains. The TE–PI controller shows slight improvement over the other IO controllers, and the IAE, ISE, and ITAE are observed to be 59.32, 3,258, and 1,858, respectively.

$$IAE = \int_0^t \|e(t)\|\, dt$$
$$ISE = \int_0^t (e(t)^2) dt \qquad (22)$$
$$ITAE = \int_0^t t \cdot \|e(t)\|\, dt$$

The speed tracking of all four controllers for a square-wave reference is compared in Fig. 7. As the reference changes from 1400 to −1400 RPM and vice versa, a step change of 2800 RPM is applied, and the percentage overshoots are observed to be 6.5% for FO–PI controller, 41.78% for both CC–PI and ZN–PI, and 27% for TE–PI controller. The FO–PI controller for the initial step of 1,400 RPM has an overshoot of 4.1%, whereas the CC–PI and ZN–PI controllers have an overshoot of 52.9%. This result shows that the overshoot of the FO–PI controller is significantly lesser than those of the CC–PI and ZN–PI controllers, thereby showing better performance of FO–PI controller compared with the CC–PI and ZN–PI controllers. The overshoot of the TE–PI controller is observed to be 10%, which is less than that of the CC–PI and ZN–PI controllers. However, the speed tracking performance of the TE–PI controller is poorer compared with that of the FO–PI controller.

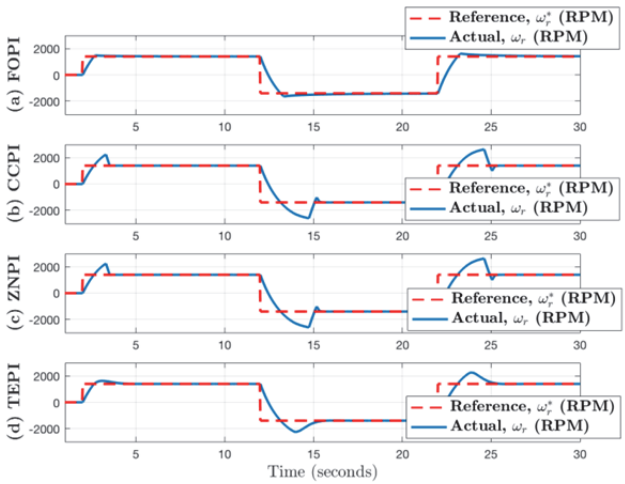

Fig. 3. Simulation results for step response of FO–PI controller at 1400 RPM: (a) FO–PI controller, (b) CC–PI controller, (c) ZN–PI controller, and (d) TE–PI controller.

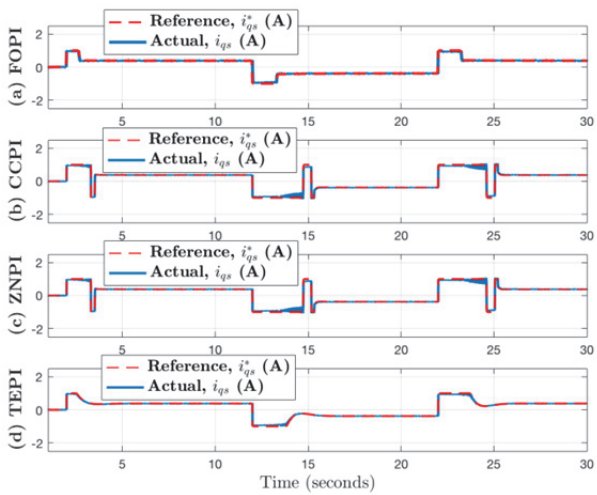

Fig. 4. Control effort during square-wave tracking at 1400 RPM: (a) FO–PI controller, (b) CC–PI controller, (c) ZN–PI controller, and (d) TE–PI controller.

TABLE II
Induction Motor Parameters

415 V	R_S = 47.5 Ω	L_{ls} = 145.5 mH, L_{lr} = 122.5 mH
0.4 A	R_r = 34.29 Ω	L_m = 750.9 mH
175 W	4 poles	1475 RPM

A = ampere, V = Volt, W = Watt, m = milli, H = Henry, RPM = Revolutions per minute, Ω = Ohm.

The performances of the four controllers are also compared in terms of the actual i_{qs} required by the controllers to track the reference. The high K_p and K_i gains of the CC–PI and ZN–PI controllers force the use of high i_{qs}, which runs the controllers into saturation. The command i^*_{qs} and actual i_{qs} for all the four controllers are plotted in Figs. 8 (a) to (d). During the transient, all controllers apply the maximum i_{qs} to track the reference. However, when the motor is running in the steady-state condition at 1400 RPM, the actual i_{qs} of the FO–PI controller is limited between 0.1606 and 0.3748 (only positive control effort). The control output of CC–PI and ZN–PI controllers [Figs. 8 (b) and (c)] constantly switches between 0.6 and −0.17 during steady-state operation at 1400 RPM. Moreover, the mean of the absolute values of i_{qs} observed in the case of the FO–PI controller is 0.2481, lesser than the absolute mean of the i_{qs} observed in the case of the CC–PI and ZN–PI controllers at 0.3028 and 0.3037, respectively. The absolute mean of the actual (i_{qs}) in case of TE–PI controller is almost the same as that of the FO–PI controller, that is, 0.2446, which shows that FO–PI and TE–PI have similar steady-state control efforts. However, the lower control effort of the TE–PI controller degrades the transient performance. This finding indicates that the designed FO–PI controller requires lesser control effort (i^*_{qs}) than the IO CC–PI and ZN–PI controllers for tracking the same reference. Thus, the FO–PI controller can maximize the torque per ampere output of the machine.

B. Low Speed Performance and Disturbance Rejection

In this section, the performances of FO–PI and IO controllers are compared for low-speed tracking and disturbance rejection. A step command of 50 RPM is selected for testing the low-speed tracking. Figs. 9 and 10 show the performances of the FO–PI, CC–PI, ZN–PI, and TE–PI controllers. All four controllers can effectively track the desired reference. However, the CC–PI and ZN–PI controllers require more control effort, that is, quadrature current (i_{qs}), than the FO–PI controller. The absolute mean value of the actual i_{qs} required by the FO–PI controller is 0.1859, lower than the absolute mean value of actual i_{qs} in the cases of CC–PI, ZN–PI, and TE–PI controllers at 0.2701, 0.2512, and 0.1818, respectively. Thus, the FO–PI controller gives better torque per ampere output compared with PI controllers in the case of low-speed tracking. However, the TE–PI controller can track the reference with lesser control effort, but the motor response to the step command has a large time delay, as shown in Fig. 9(d).

Next, we investigate the disturbance rejection capability of the controllers by adding a step load at 10 seconds, followed by its removal at 20 seconds while the motor is running at 1400 RPM. The speed response and the required control effort (i_{qs}) are plotted in Figs. 11 and 12, respectively. The speed response of the FO–PI controller shows that the controller can completely reject the disturbance. The CC–PI and ZN–PI controllers also reject the external disturbance by applying a much higher actual i_{qs} than the FO–PI controller. The FO–PI controller applies an absolute mean i_{qs} of 0.633, which is less than the control effort of 0.6680 and 0.7050 required by the CC–PI and ZN–PI controllers, respectively. The TE–PI can also reject the disturbance with a control effort (0.635) almost like that of the FO–PI controller, as shown in Fig. 12. However, the motor speed has a dip of about 200 RPM when the load is added and an overshoot of 200 RPM upon load removal. Thus, the CC–PI and ZN–PI controllers require a high control effort for the external

TABLE III
DRIVE SYSTEM SPECIFICATIONS

Inverter rating	500 V DC, 10 KHz
Motor rating	415 V, 175 W
Controller	dSPACE 1103
Sampling time	0.1 ms
Current sensor	25 A (LEM LA-25-NP)
Encoder	360 pulses per revolution

A = ampere, V = Volt, W = Watt, m = milli, H = Henry,
RPM = Revolutions per minute, Ω = Ohm.

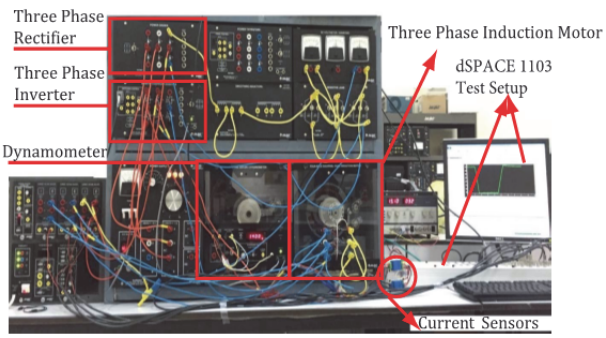

Fig. 5. Experimental setup for the IFO drive system.

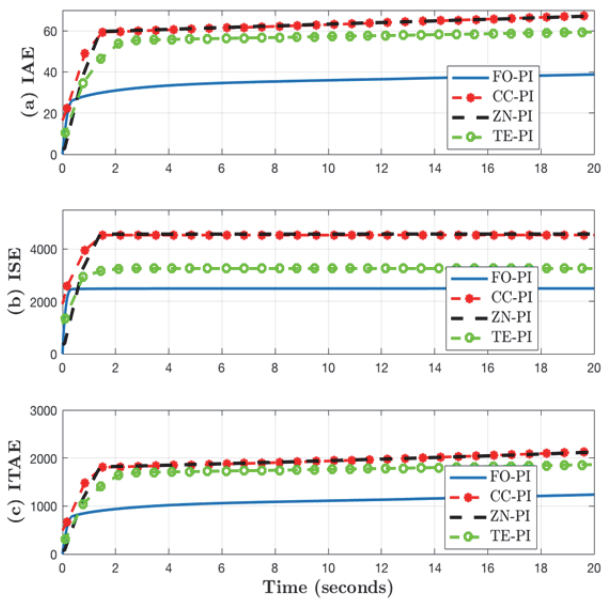

Fig. 6. Comparison of step response performance of all controllers.

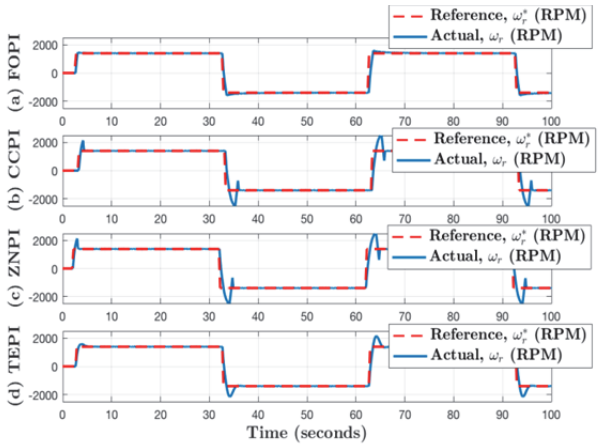

Fig. 7. Performance comparison of FO–PI and PI controllers for 1400 RPM square-wave speed tracking: (a) FO–PI controller, (b) CC–PI controller, (c) ZN–PI controller, and (d) TE–PI controller. The red curve shows the speed command, while the blue curve represents the actual speed of the motor.

disturbance rejection, whereas the TE–PI controller is unable to effectively reject the external disturbance.

C. Controller's Performance Under Detuning

The IFO control of an induction machine depends upon the feed forward slip calculation for which the knowledge of accurate rotor time constant is critical. However, heating and saturation effects can cause variations in the rotor time constant. If the rotor time constant varies and its value is not adapted in real time, it will result in a wrong slip command and loss of field orientation, which degrades machine performance. The drive system when fed with a wrong rotor time constant is termed as detuned. The two solutions to overcome the detuning problem are fixing the rotor time constant by parameter adaptation or designing a controller that is less sensitive to the variations in rotor time constant. This work evaluates the performance of FO–PI for a detuned IFO induction motor drive system and compares its performance with that of PI controllers under the detuned condition. The rotor time constant $T_r = L_r/R_r$ of the prototype induction motor drive system can be calculated using the motor parameters given in Table II. Its value is found to be 46.5879×10^{-3}. Thus, the value of $1/T_r$ used in the field-orientation block for slip calculation is 21.46. The drive system is operated at 1400 RPM, and the reciprocal of the rotor time constant ($1/T_r$) is initially doubled (42.92) and then brought to half (10.73) of its nominal value, as shown in Fig. 13(a). A load of 0.2 Nm torque is likewise added to rigorously compare the controllers' performance under the detuned operating condition. The speed tracking of the FO–PI controller is almost unaffected when $1/T_r$ is doubled and is reduced to half of its nominal value. However, an overshoot of 20 RPM is observed as the $1/T_r$ is brought back from half to its normal value at 58 seconds. The CC–PI and ZN–PI

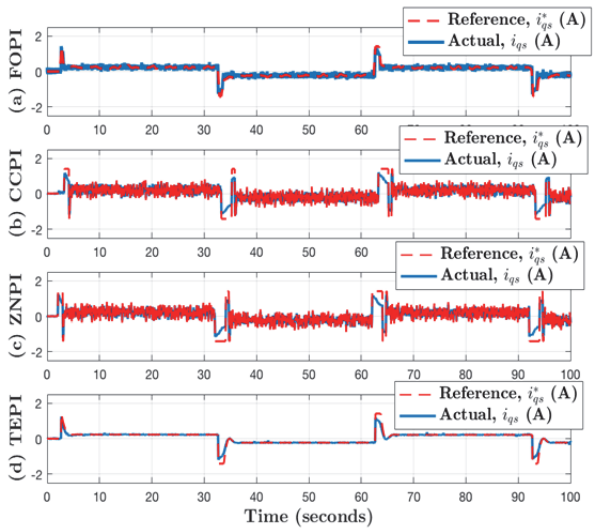

Fig. 8. Control effort comparison of FO–PI and PI controllers for 1400 RPM square-wave speed tracking: (a) FO–PI controller, (b) CC–PI controller, (c) ZN–PI controller, and (d) TE–PI controller. The red curve shows the controllers' torque command current, while the blue curve represents the actual current.

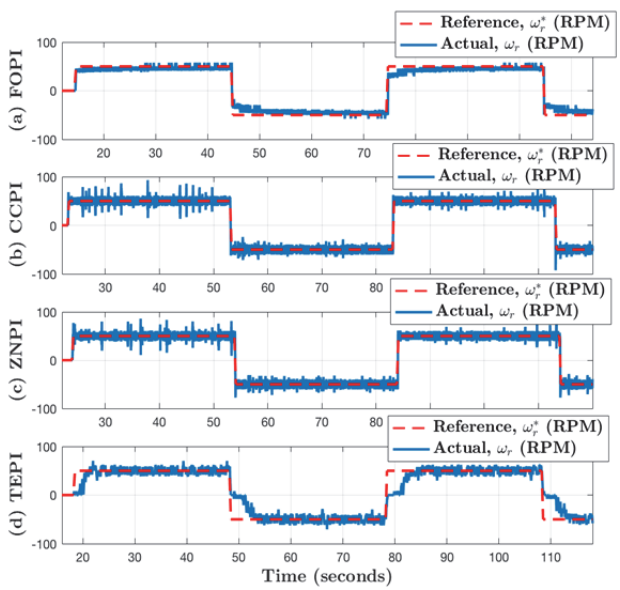

Fig. 9. Performance comparison of FO–PI and PI controllers for low speed tracking at 50 RPM: (a) FO–PI controller, (b) CC–PI controller, (c) ZN–PI controller, and (d) TE–PI controller. The red curve shows the speed command, while the blue curve represents the actual speed of the motor.

controllers can also maintain the speed, but they start to destabilize and lose the speed tracking as the detuning persists for a longer period. High overshoot is observed in the case of the TE–PI controller every time the $1/T_r$ is changed. Thus, the FO–PI controller can track the motor speed better than the CC–PI, ZN–PI, and TE–PI controllers.

The actual i_{qs} of FO–PI and PI controllers under the detuned operation are evaluated by plotting the required actual torque current i_{qs} in Fig. 14. The controller effort for

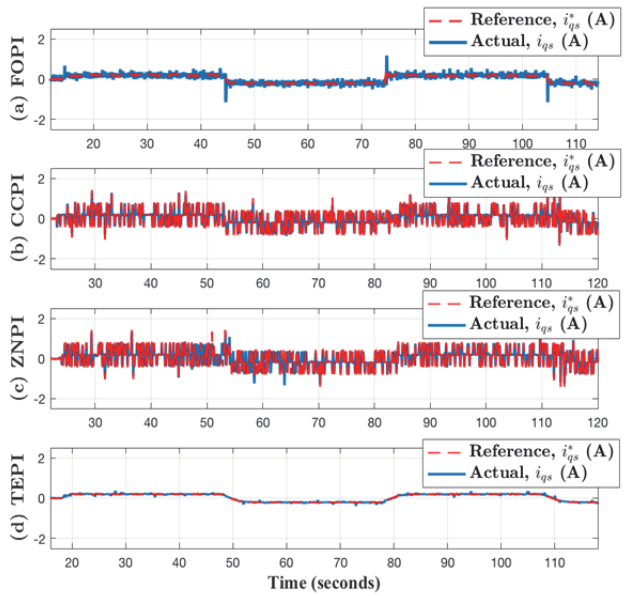

Fig. 10. Control effort comparison of FO–PI and PI controllers during low speed tracking at 50 RPM: (a) FO–PI controller, (b) CC–PI controller, (c) ZN–PI controller, and (d) TE–PI controller. The red curve shows the controllers' torque command current, while the blue curve represents the actual current.

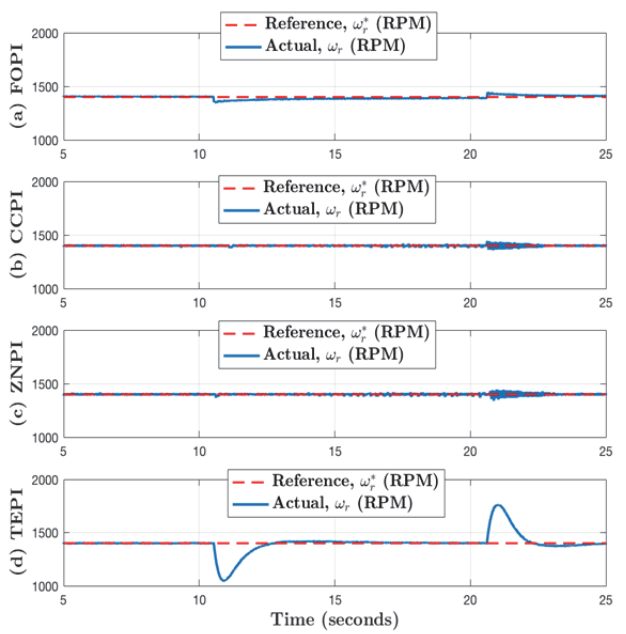

Fig. 11. Disturbance rejection comparison of FO–PI and PI controllers while operating at 1400 RPM: (a) FO–PI controller, (b) CC–PI controller, (c) ZN–PI controller, and (d) TE–PI controller. The red curve shows the speed command, while the blue curve represents the actual speed of the motor.

the FO–PI controller shown in Fig. 14(a) shows a slight increase to overcome the detuning effects. In the case of the FO–PI controller, the i_{qs} varies between the maximum value of 0.53 and the minimum value of 0.26. This range increases when the machine is controlled by the CC–PI and ZN–PI

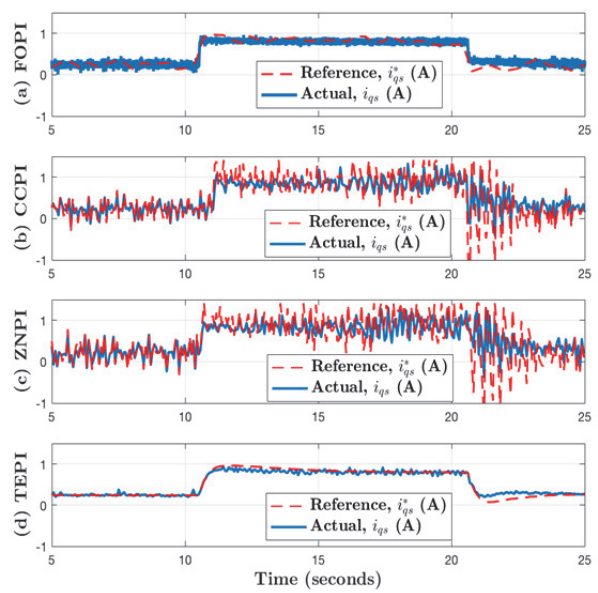

Fig. 12. Control effort comparison of FO–PI and PI controllers during disturbance rejection test at 1400 RPM: (a) FO–PI controller, (b) CC–PI controller, (c) ZN–PI controller, and (d) TE–PI controller. The red curve shows the controllers' torque command current, while the blue curve represents the actual current.

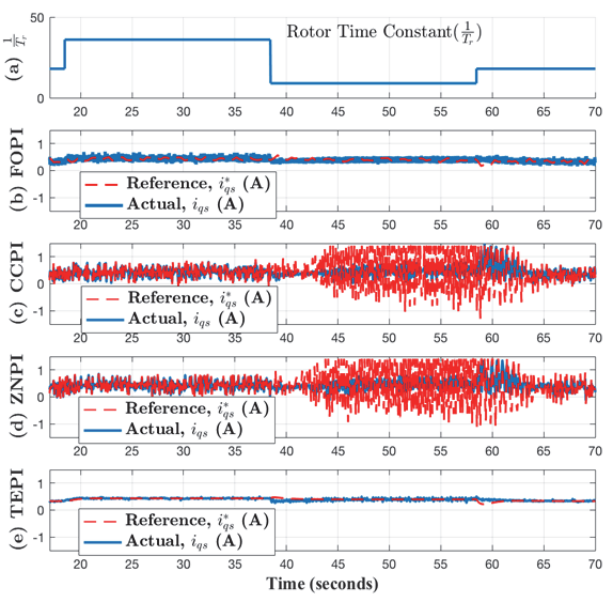

Fig. 13. Performance comparison of FO–PI and PI controllers for a detuned machine operating at 1400 RPM: (a) Variation in rotor time constant (1/Tr), (b) FO–PI controller, (c) CC–PI controller, (d) ZN–PI controller, and (e) TE–PI controller. The red curve shows the speed command, while the blue curve represents the actual speed of the motor. (The red–blue distinction is not valid for the first curve.)

controllers between 0.836 and –0.05. The i_{qs} required by the TE–PI controller varies between 0.37 and 0.49, lower than the other three controllers but at the expense of a high overshoot.

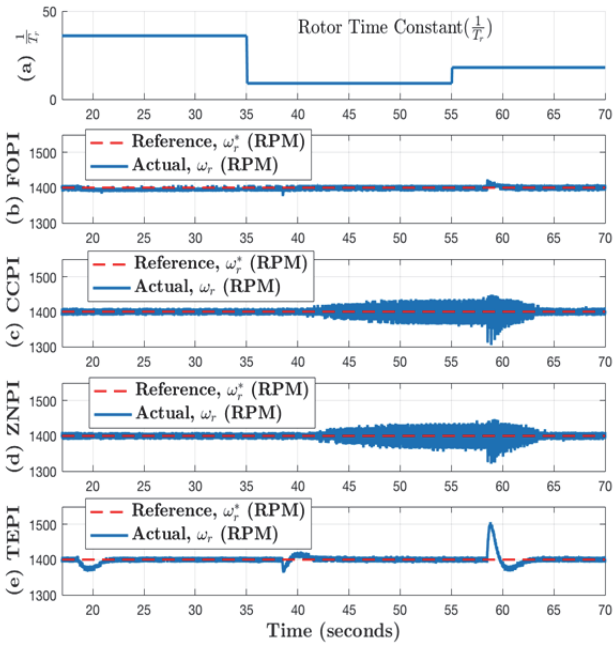

Fig. 14. Control effort comparison of FO–PI and PI controllers for a detuned machine operating at 1400 RPM: (a) Variation in rotor time constant (1/Tr), (b) FO–PI controller, (c) CC–PI controller, (d) ZN–PI controller, and (e) TE–PI controller. The red curve shows the controllers' torque command current, while the blue curve represents the actual current. (The red–blue distinction is not valid for the first curve.)

Observing Figs. 14(c) and (d) between 42 and approximately 62 seconds shows that CC–PI and ZN–PI controllers are destabilized for a detuned drive system; oscillations in the motor speed around the desired reference are also observed in Figs. 13(c) and (d). Moreover, audible motor vibrations are observed in the laboratory. These vibrations can be explained by the windup of the integrator because the controller is able to return to stable operating conditions after some time, as shown in Figs. 13(c) and (d) and 14(c) and (d). The windup problem is not observed when the motor is controlled using FO–PI or TE–PI controllers.

D. Controller's Performance in Field-weakening Region

Motor flux in the field-weakening region must be reduced to decrease the motor back EMF and increase the motor speed. Therefore, below the base speed, the flux command current i^*_{ds} is fixed at 0.4 A, and above the base speed field, weakening is applied using Eq. (23) while motor speed is linearly increased up to 2500 RPM. The results plotted in Figs. 15 and 16 show the drive system performance as the speed is increased linearly from 600 RPM to 2500 RPM. The flux command current i^*_{ds} is plotted in both Figs. 15 (a) and 16 (a), and it reaches a final value of 0.2 A at 2500 RPM.

$$i^*_{ds} = \begin{cases} 0.4, & \omega_r < 1500 \\ 0.00019\omega_r + 0.68, & 1500 \leq \omega_r \leq 2500 \end{cases} \quad (23)$$

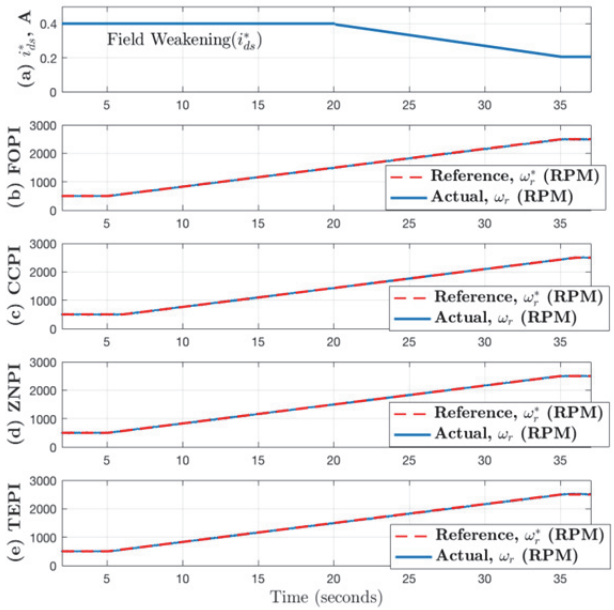

Fig. 15. Performance comparison of FO–PI and PI controllers in the field-weakening region while speed changes from 600 RPM to 2500 RPM: (a) i^*_{ds}, (b) FO–PI controller, (c) CC–PI controller, (d) ZN–PI controller, and (e) TE–PI controller. The red curve shows the speed command, while the blue curve represents the actual speed of the motor.

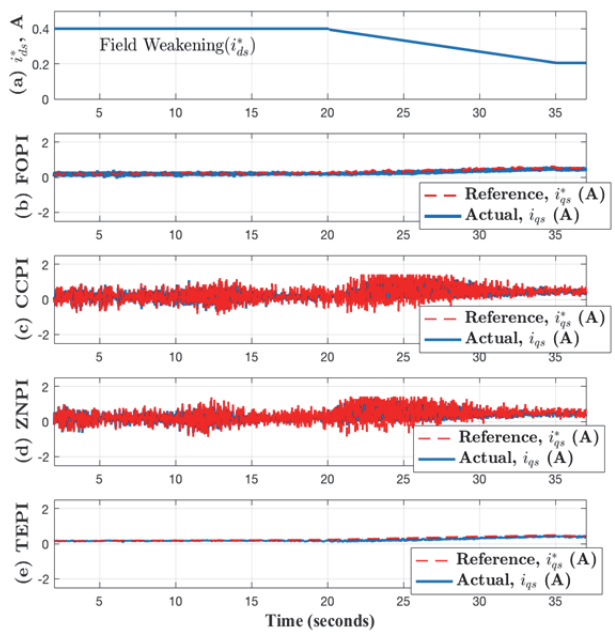

Fig. 16. Control effort comparison of FO–PI and PI controllers in the field-weakening region while speed changes from 600 RPM to 2500 RPM: (a) i^*_{ds}, (b) FO–PI controller, (c) CC–PI controller, (d) ZN–PI controller, and (e) TE–PI controller. The red curve shows the controllers' torque command current, while the blue curve represents the actual current.

The command and actual control efforts i^*_{qs} and i_{qs} are plotted in Fig. 16 for all controllers. The mean i_{qs} required by the FO–PI controller is 0.3131 A, which is considerably

lesser than the ones required by CC–PI and ZN–PI controllers at 0.3323 and 0.3286 A, respectively. The TE–PI controller performs slightly better than the FO–PI controller, with a mean torque control effort of 0.2158 A.

V. CONCLUSIONS

An FPDT model of the induction motor is developed for the speed control of an IFO drive system, after which FO–PI and IO–PI (CC–PI, ZN–PI, and TE–PI) controllers are designed. A quantitative comparison using IAE, ISE, and ITAE shows that the FO–PI controller performs better than the ZN–PI, CC–PI, and TE–PI controllers. The three integer controllers resulted in a high overshoot both in simulation and experimentation when the drive system is tested for square-wave tracking. However, the FO–PI controller shows comparatively negligible overshoot both in the simulation and experimentation. The torque command current required by the FO–PI controller also shows that it provides better torque per ampere output of the machine compared with linear PI controllers.

Moreover, the FO–PI controller demonstrates good disturbance rejection characteristics with minimal control effort compared with the IO controllers and exhibits better speed tracking for a detuned IFO drive system. By contrast, the speed tracking is degraded and an unstable motor operation is observed when ZN–PI and CC–PI controllers are used. Finally, the FO–PI controller shows good performance in the field-weakening region of the drive system and provides higher torque per ampere output of the machine.

ACKNOWLEDGMENT

The authors would like to thank the Office of Research and Graduate Studies at the American University of Sharjah, Sharjah, UAE for funding this work through the research grant FRG14-2-25.

REFERENCES

[1] J. G. Ziegler and N. B. Nichols, "Optimum settings for automatic controllers," *Trans. ASME*, Vol. 64, No. 11, 1942.

[2] G. Cohen and G. Coon, "Theoretical consideration of retarded control," *Trans. ASME*, Vol. 75, No. 1, pp. 827-834, 1953.

[3] J. Han, "From PID to Active Disturbance Rejection Control," *IEEE Trans. Industrial Electron.*, Vol. 56, No. 3, pp. 900-906, Mar. 2009.

[4] T. Terras, S. Hadjeri, A. Mezouar, and T. M. Chikouche, "Robust speed control with rotor resistance estimation," *Canadian J. Elect. Comput. Eng.*, Vol. 36, No. 2, pp. 43-51, 2013.

[5] G. Feng, Y.-F. Liu, and L. Huang, "A new robust algorithm to improve the dynamic performance on the speed control of induction motor drive," *IEEE Trans. Power Electron*., Vol. 19, No. 6, pp. 1614-1627, Nov. 2004.

[6] A. Alexandridis, G. Konstantopoulos, and Q. Zhong, "Advanced integrated modeling and analysis for adjustable speed drives of induction motors operating with minimum losses," *IEEE Trans. Energy Convers*., Vol. 30, No. 3, pp. 1237-1246, Sep. 2015.

[7] Z. Xu, J. Wang, and P. Wang, "Passivity-based control of induction motor based on Euler-Lagrange (EL) model with flexible damping," in *Int. Conf. Elect. Machines and Syst*., pp. 48-52, 2008.

[8] S. Chai, L. Wang, and E. Rogers, "A cascade MPC control structure for a PMSM with speed ripple minimization," *IEEE Trans. Ind. Electron*., Vol. 60, No. 8, pp. 2978-2987, Aug. 2013.

[9] J. Li, H.-P. Ren, and Y.-R. Zhong, "Robust speed control of induction motor drives using first-order auto-disturbance rejection controllers," *IEEE Trans. Ind. Appl*., Vol. 51, No. 1, pp. 712-720, Jan. 2015.

[10] E. Fuentes, D. Kalise, J. Rodriguez, and R. Kennel, "Cascade-free predictive speed control for electrical drives," *IEEE Trans. Ind. Electron*., Vol. 61, No. 5, pp. 2176-2184, May 2014.

[11] P. Alkorta, O. Barambones, J. Cortajarena, and A. Zubizarrreta, "Efficient multivariable generalized predictive control for sensorless induction motor drives," *IEEE Trans. Ind. Electron*., Vol. 61, No. 9, pp. 5126-5134, Sep. 2014.

[12] H. Michalska and D. Mayne, "Receding horizon control of nonlinear systems," in *Proc. 28th IEEE Conf. Decision Control*, Vol. 1, pp. 107-108, 1989.

[13] V. Utkin, "Sliding mode control design principles and applications to electric drives," *IEEE Trans. Ind. Electron*., Vol. 40, No. 1, pp. 23-36, Feb. 1993.

[14] X. Zhang, "Sensorless induction motor drive using indirect vector controller and sliding-mode observer for electric vehicles," *IEEE Trans. Veh. Technol*., Vol. 62, No. 7, pp. 3010-3018, Sep. 2013.

[15] Y. Wang, Z. Wang, J. Yang, and R. Pei, "Speed regulation of induction motor using sliding mode control scheme," in *Conf. Rec. IEEE-IAS Annu. Meeting*, Vol. 1, pp. 72-76, 2005.

[16] A. Saghafinia, H. W. Ping, M. Uddin, and K. Gaeid, "Adaptive fuzzy sliding-mode control into chattering-free IM drive," *IEEE Trans. Ind. Appl*., Vol. 51, No. 1, pp. 692-701, Jan. 2015.

[17] B. Heber, L. Xu, and Y. Tang, "Fuzzy logic enhanced speed control of an indirect field-oriented induction machine drive," *IEEE Trans. Power Electron*., Vol. 12, No. 5, pp. 772-778, Sep. 1997.

[18] M. Uddin, Z. R. Huang, and A. Hossain, "Development and implementation of a simplified self-tuned neuro fuzzy-based IM drive," *IEEE Trans. Ind. Appl*., Vol. 50, No. 1, pp. 51-59, Jan. 2014.

[19] M. Masiala, B. Vafakhah, J. Salmon, and A. Knight, "Fuzzy self-tuning speed control of an indirect field-oriented control induction motor drive," *IEEE Trans. Ind. Appl*., Vol. 44, No. 6, pp. 1732-1740, Nov. 2008.

[20] S. K. Sahoo, T. Bhattacharya, and M. Aravind, "A synchronized sinusoidal PWM based rotor flux oriented controlled induction motor drive for traction application," in *Proc. IEEE APEC*, pp. 797-804, 2013.

[21] P.-Y. Lin and Y.-S. Lai, "Novel voltage trajectory control for field weakening operation of induction motor drives," *IEEE Trans. Ind. Appl.*, Vol. 47, No. 1, pp. 122-127, Jan./ Feb. 2011.

[22] S.-H. Kim and S.-K. Sul, "Maximum torque control of an induction machine in the field weakening region," *IEEE Trans. Ind. Appl*., Vol. 31, No. 4, pp. 787-794, Jul./Aug. 1995.
[23] M. Mengoni, L. Zarri, A. Tani, G. Serra, and D. Casadei, "A comparison of four robust control schemes for field-weakening operation of induction motors," *IEEE Trans. Power Electron*., Vol. 27, No. 1, pp. 307-320, Jan. 2012.
[24] D. Valerio, "Ninteger v. 2.3 fractional control toolbox for Matlab," 2005. [Online]. Available: https://www.mathworks.com/matlabcentral/fileexchange/8312-ninteger
[25] A. Khurram, "Performance enhancement of field oriented induction motor drive system," Master's Thesis, American University of Sharjah, 2016.
[26] A. Khurram, H. Rehman, and S. Mukhopadhyay, "A high performance speed regulator design for ac machines," in *Applied Power Electronics Conference and Exposition (APEC)*, pp. 2782-2787, 2016.
[27] Y. Chen, T. Bhaskaran, and D. Xue, "Practical tuning rule development for fractional order proportional and integral controllers," *J. Computational Nonlinear Dynamics*, Vol. 3, No. 2, p. 021403, Feb. 2008.
[28] T. Hägglund and K. J. Åstrom, "Revisiting the Ziegler-Nichols tuning rules for PI control," *Asian J. Contr*., Vol. 4, No. 4, pp. 364-380, 2002.
[29] S. Mukhopadhyay, "Fractional order modeling and control: development of analog strategies for plasma position control of the STOR-1M tokamak," Master's Thesis, Utah State University, p. 460, 2009.
[30] P. Lanusse, J. Sabatier, and A. Oustaloup, "Extension of PID to fractional orders controllers: a frequency-domain tutorial presentation," *IFAC Proceedings*, Vol. 47, No. 3, pp. 7436-7442, 2014.
[31] S. Manabe, "The non-integer integral and its application to control systems," *Japanese Inst. Electrical Engineers J.*, Vol. 80, No. 860, pp. 589-597, 1960.
[32] I. Podlubny, *Fractional Differential Equations: An Introduction to Fractional Derivatives, Fractional Differential Equations, to Methods of Their Solution and Some of Their Applications*, Academic Press, Vol. 198, 1998.

Adil Khurram received his undergraduate degree in Electrical Engineering from the National University of Sciences and Technology, Islamabad, Pakistan in 2013 and his M.Sc. in Electrical Engineering from the American University of Sharjah (AUS), Sharjah, UAE in 2016. During his masters, he worked on field-oriented control of three-phase induction motors and developed fractional-order controllers for the speed control of induction motors. He is currently pursuing a Ph.D. degree in Electrical Engineering from the University of Vermont (UVM), Vermont, USA. His research interests include aggregate modeling and control of thermostatically controlled loads and batteries for direct load control in power systems.

Habibur Rehman received his B.Sc. degree in Electrical Engineering from the University of Engineering and Technology Lahore, Pakistan, in 1990. He received his M.S. and Ph.D. degrees in Electrical Engineering from the Ohio State University, Columbus, Ohio in 1995 and 2001, respectively. He has a wide experience in the area of power electronics and motor drives in both industry and academia. From July 1998 to December 1999, he worked as a design engineer in the Ecostar Electric Drives and Ford Research Laboratory, where he was a member of the Electric, Hybrid, and Fuel Cell vehicle development programs. From 2001 to 2006, he worked in the Department of Electrical Engineering at the United Arab Emirates (UAE) University, Al-Ain, UAE as an assistant professor. In 2006, he joined the Department of Electrical Engineering at the American University of Sharjah, where he is currently working as an associate professor. Dr. Rehman is the recipient of the Best Teacher award from the College of Engineering, UAE University for the academic year 2002/2003. His primary research interests are in the areas of power electronics and their application to power systems, adjustable-speed drives, and alternative energy vehicles.

Shayok Mukhopadhyay received his B.E. from the College of Engineering Pune (C.O.E.P.), University of Pune, India, 2006. He received his M.Sc. from Utah State University, Logan, UT, USA, 2009 and his Ph.D. from the Georgia Institute of Technology, Atlanta, GA, USA, 2014. All the above degrees are in Electrical Engineering. He has been an assistant professor in the Department of Electrical Engineering at the American University of Sharjah, UAE since 2014. His research interests include nonlinear systems, computational methods, battery modeling and failure detection, and robotic path planning. He received an award for best presentation in the "Nonlinear Systems III" session in the American Control Conference (ACC) 2014. He was part of a five-person team that won the national category of the UAE AI & Robotics for Good Awards in 2017 for developing an "In-Pipe Inspection Robot."

Daniyal Ali received his bachelor degree in Electrical Engineering from the National University of Sciences and Technology (NUST), Pakistan in 2013. He completed his Masters at the American University of Sharjah (AUS), UAE in 2016. Presently, he is working at Sky Electric as a research and development engineer on smart energy storage systems. Previously, he served as a lecturer at NUST, Pakistan for one year.