JPE 16-1-22

http://dx.doi.org/10.6113/JPE.2016.16.1.205
ISSN(Print): 1598-2092 / ISSN(Online): 2093-4718

Intelligent Position Control of a Vertical Rotating Single Arm Robot Using BLDC Servo Drive

Manikandan R.† and Arulmozhiyal R.*

†,*Department of Electrical and Electronics Engineering, Sona College of Technology, Tamilnadu, India

Abstract

The manufacturing sector resorts to automation to increase production and homogeneity of products during mass production, without increasing scarce, expensive, and unreliable manpower. Automation in the form of multiple robotic arms that handle materials in all directions in different stages of the process is proven to be the best way to increase production. This paper thoroughly investigates robotic single-arm movements, that is, 360° vertical rotation, with the help of a brushless DC motor, controlled by a fuzzy proportional–integral–derivative (PID) controller. This paper also deals with the design and performance of the fuzzy-based PID controller used to control vertical movement against the limited scope of conventional PID feedback controller and how the torque of the arm is affected by the fuzzy PID controller in the four quadrants to ensure constant speed and accident-free operation despite the influence of gravitational force. The design was simulated through MATLAB/SIMULINK and integrated with dSPACE DS1104-based hardware to verify the dynamic behaviors of the arm.

Key words: Brushless direct-current (BLDC) motor, dSPACE DS1104, Fuzzy proportional–integral–derivative (PID) controller, Position servo drive, Vertical rotating single arm robot

I. INTRODUCTION

Brushless direct-current (BLDC) motors are widely used in servo drive applications. These motors have less inertia and are therefore suitable for swift start and stop operations. BLDC motors are now available from milliwatt to kilowatt for operating robotic arms in the automotive, aerospace, transport, and medical fields, as well as in many other industrial automation applications [1]-[3]. Industrial automation involves operations in different environments with different payloads at a uniform speed in the four quadrants with bidirectional speed controls and regenerative braking capabilities [4]-[7]. The conventional feedback proportional–integral–derivative (PID) controller is based on the accuracy of the mathematical model of the system, and its expected performance is likely to be affected by load disturbances, whereas the latest fuzzy logic is a dependable technique and achieves better dynamic performance with least chances of errors. This paper discusses the position control drive of a vertically rotating single-arm robot using conventional PID and fuzzy PID controllers.

This paper is organized as follows: The first section presents the introduction. The second section discusses the structure of a vertically rotating arm and a position servo drive. The third section explains the simulation carried through MATLAB/SIMULINK. The fourth section elaborates the hardware implementation through a dSPACE environment. The fifth section describes the designs of conventional PID and fuzzy PID controllers. The sixth section consists of detailed results and discussion about the various movements of the robotic arm in the four quadrants, showing that fuzzy PID controller is more reliable than the conventional PID controller. The last section presents the conclusion.

II. POSITION SERVO DRIVE OF A SINGLE-ARM ROBOT

The single arm of a robot moves vertically or horizontally with one degree of freedom (DOF) around 360°. During its horizontal motion, no change occurs in the torque of the payload because gravitational force acts on the center of the arm at a constant magnitude. However, during vertical motion, payload torque varies depending on its position

Manuscript received Jun. 15, 2015; accepted Sep. 29, 2015
Recommended for publication by Associate Editor Gaolin Wang.
†Corresponding Author: electricmani@yahoo.co.in
Tel: +91-9944270473, Sona College of Technology, India
*Department of Electrical and Electronics Engineering, Sona College of Technology, India

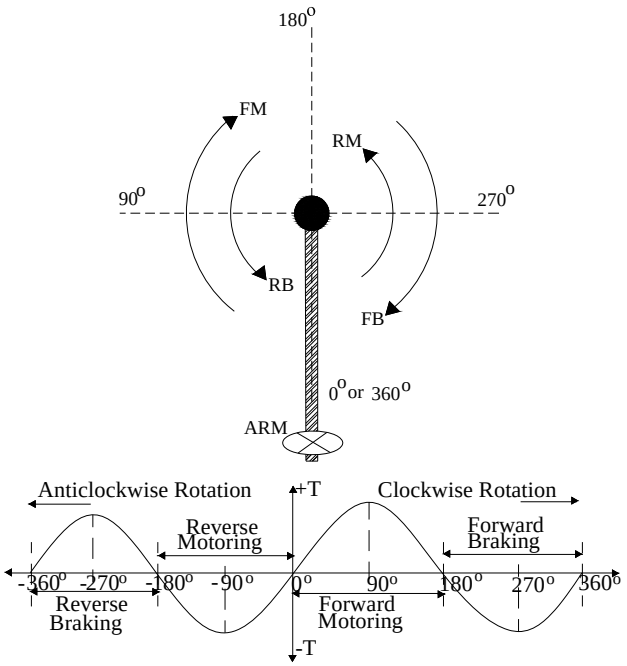


Fig. 1. Single-link vertical rotating arm and torque with effect of gravity as a function of angle.

because of the gravitational force in the centric load [8]. Fig. 1 shows the arm rotating vertically in 360° schematic and the relevant load torque curve.

When the arm moves from 0° to 180° in a clockwise direction, the motor rotates in a forward direction. The arm moves at a staggered speed of nearly 90°, and the load torque at 90° is the maximum. The constant speed of the motor has to be maintained without reference to the load torque variations to ensure that the arm reaches the desired destination without any delay. When the arm rotates downward in a clockwise direction at 180°–360°, the gravitational force pulls down the arm. The speed and current are likely to exceed the safe/rated limit. The arm will also reach the destination faster than the motoring mode. The load torque during this operation is negative, and the motor rotates in a forward direction. The negative torque of the motor operates the forward braking (FB) mode to prevent sudden drop of payload, which causes accidents and damages.

Similarly, when the arm rotates in an anticlockwise direction, the same scenario appears. When the robotic arm rotates in an anticlockwise direction at 0°– -180° (360°–180°), the motor runs in the negative direction to lift the payload and gives negative torque, which causes the motor to operate in a reverse motoring (RM) mode. During the anticlockwise rotation of arm at -180°– -360° (180°–0°), the load torque will act on a positive direction. However, the motor rotates in a reverse direction and operates in the reverse braking (RB) mode. The curve that connects various positions of the arm and load torques resembles a sine waveform in both directions. Hence, the load applied to the motor shaft (T_m) is defined as the trigonometric sin function of actual position multiplied with payload torque (T_L) as follows:

$$T_m = T_L \sin(\theta) \quad (1)$$

Where T_m is the mechanical shaft torque given to the motor in n–m, T_L is the payload toque added with arm in n–m, and θ is the arm position in degrees.

The block diagram of the proposed position servo drive is shown in Fig. 2. The single arm is directly coupled with the motor shaft and a quadrature encoder, which is used to measure the actual position. Quadrature encoder pulse (QEP) signals Qa and Qb are decoded through a quadrature pulse decoder to obtain the information about the exact position of the arm. The actual and set positions of the arm are compared to obtain the details of the position error, and these details are fed to the position controller [9]–[11], which is either conventional PID or fuzzy PID model. The output of the position controller gives the magnitude of reference DC link current ±Idc*, which is used within the safety range/limit; the sign indicates the direction where the motor should run. The absolute reference |±Idc*| current is compared with the actual current measured as Idc to arrive at the current error to feed the conventional or fuzzy current controller.

The output of current controller decides the duty cycle (0–1) of the gate pulse. The current controller will act in the inner loop, and the position controller will act in the outer loop of the control system. The proposed position–current loop structure reduces complexity. The conventional or fuzzy-based PID position controllers produce a negative control signal to decelerate the drive before the position reaches the set point consequences in good dynamic performance.

The BLDC motor is self-controlled through hall position information. Hall signals are decoded to produce gate pulse through a hall decoder. These signals are modulated with duty cycle information received from the current controller to produce six pulse width modulated (PWM) pulses. These PWM pulses are fed to six metal–oxide–semiconductor field-effect transistors (MOSFET) or insulated-gate bipolar transistor (IGBT) switch-based inverters to drive a three-phase BLDC motor. Available AC supply is converted to constant DC through an uncontrolled diode rectifier.

Fig. 3 shows the four modes of operating quadrants. When the arm moves upward, both in clockwise and anticlockwise directions, the speed and torque signs are the same but are in contrast with each other. When the arm moves upward in the clockwise direction, the motor operates in the first quadrant of the forward motoring (FM) mode. When it runs upward in the anticlockwise direction, the motor operates in the third quadrant of the RM mode. When the arm rotates downward in both directions, the speed and torque signs are opposite each other, but when these signs are added together, the motor can work on the braking mode. Subsequently, when the arm moves in the clockwise downward direction, the

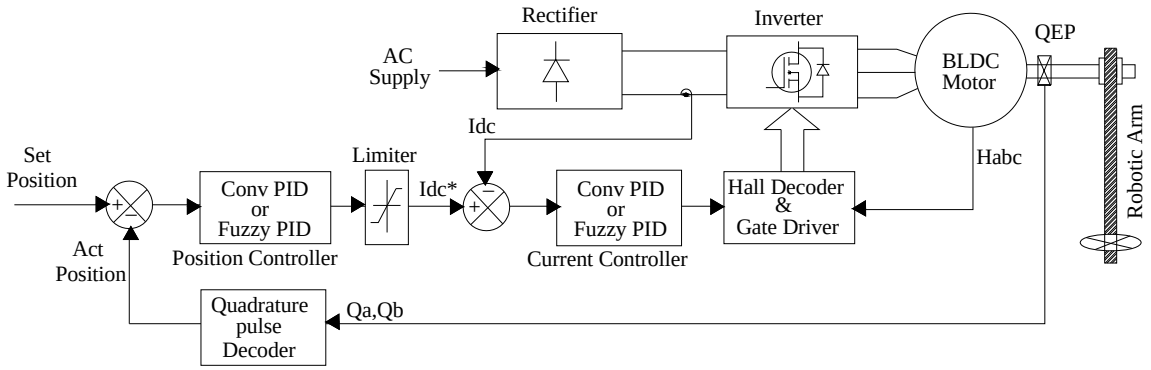


Fig. 2. Block diagram of the arm-position servo drive.

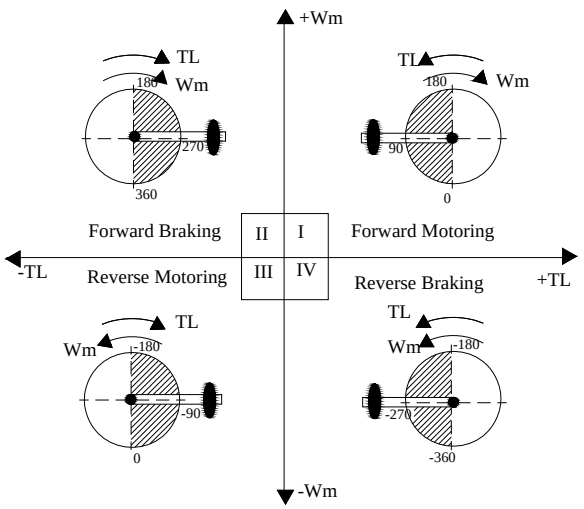


Fig. 3. Four quadrant modes.

TABLE I
MOTOR SPECIFICATIONS

Motor Power Company, Italy. TETRA 85TR2.2	
Voltage	310V dc
Speed	4600 rpm
Torque	2.2 Nm
Stall current	4.52 A
Poles number	4
Voltage constant (Ke)	51 V/Krpm
Torque constant (Kt)	0.49 Nm/A
Ph/ph resistance	3.07 Ω
Ph/ph inductance	6.57 mH
Moment of inertia (Jm)	1.8 Kg cm^2
Encoder	2000 ppr

motor will work on the second quadrant of the FB mode. Likewise, when the arm moves in the anticlockwise downward direction, the motor will work on the fourth quadrant in the RB mode. The uniform speed of the arm movements, irrespective of varying payload torques in all the four quadrants, must be maintained to reach the desired destination. Hence, the design of the feedback controller plays a vital role in these operations.

III. SIMULATION

The complete simulation model of the single-arm position servo drive through MATLAB/SIMULINK is depicted in Fig. 4. Permanent magnet synchronous motor with trapezoidal back electromotive force is modeled as a BLDC motor. TETRA 85TR2.2 series BLDC motor from Motor Power Company Italy is used with the specifications shown in Table I. Three-phase six–MOSFET-based inverters supply the three phase windings of the BLDC motor. From the hall position signals Ha, Hb, and Hc, six gate driving pulses are decoded through the gate decoder subsystem. This subsystem includes direction changing logic and the modulating unit for changing the pulse width as per duty cycle information coming from the current controller. A fixed frequency of 10 KHz chopping rate is used for pulse width modulation.

A shaft quadrature encoder subsystem model is shown in Fig. 5. The incremental shaft encoder with 2000 pulses per revolution (ppr) sensor block is used from SIMSCAPE block sets to implement the quadrature encoder; hence, connecting the solver and linking blocks with the other block sets used from SIMPOWER SYSTEM tools is necessary. This connection will generate quadrature encoder pulses Qa and Qb, and their phase shifting positions change according to the change in the direction of the rotation. Fig. 6 shows the quadrature pulse decoder subsystem, which is used to compute the position-equivalent pulse counts and direction of rotation and speed of motor [12] as

$$N_a = \left(\frac{60}{2000}\right) \times f_{Qa} \tag{2}$$

Where N_a is the motor speed in rpm, and f_{Qa} is the frequency of QEP signal Qa in hertz. Pulse frequency is converted into speed and multiplied with direction to know the directional speed in rpm as per Equ. (2). Speed is measured to show the performance but not used anywhere in the control systems. One quadrature encoder signal has 2000 falling and rising edges per revolution.

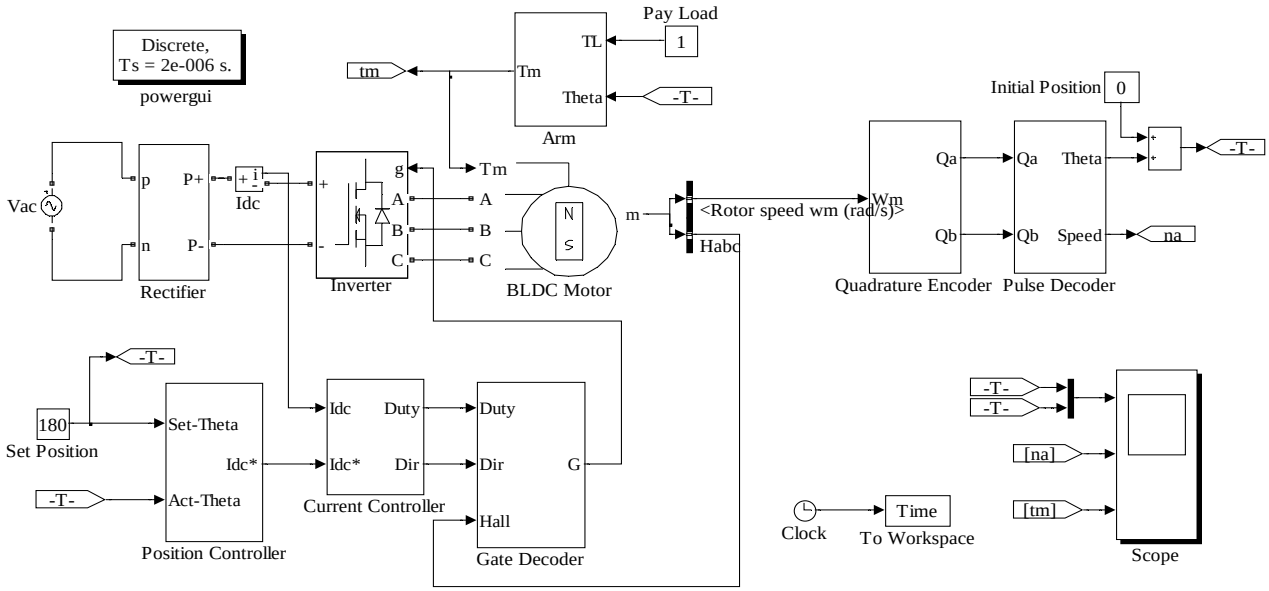


Fig. 4. SIMULINK model of vertical rotating single-arm position servo drive.

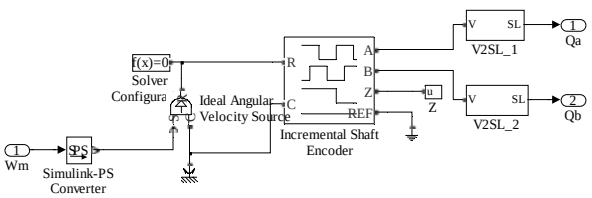


Fig. 5. Quadrature encoder subsystem.

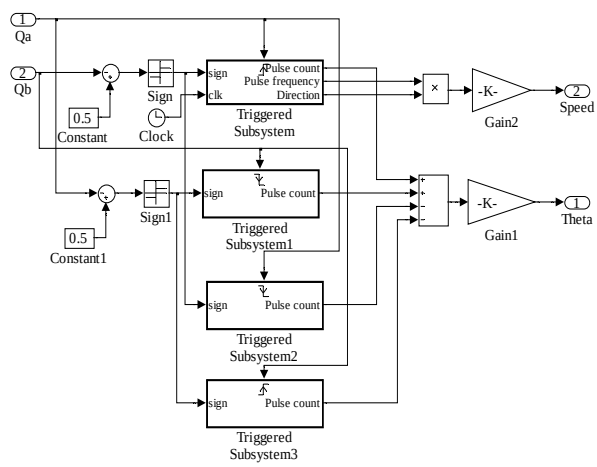


Fig. 6. Quadrature pulse decoder subsystem.

From this revolution, 8000 pulse counts are possible for one full 360° rotation of the arm. This rotation is divided into one revolution maximum pulse counts to achieve 360/8000=0.045° of a high-resolution position servo drive system. A vertically rotating single arm is modeled in the arm subsystem. The sign and magnitude of load torque is computed from sin function of actual position as per Equ. (1). External payload torque T_L that is added to the arm tip is multiplied with the function on sine to obtain mechanical load torque T_m, and the same is applied to the motor.

IV. HARDWARE IMPLEMENTATION

The laboratory hardware test bench is developed through the dSPACE DS1104 DSP controller with the same motor specification in Table I, as shown in the simulation. The vertically rotating single-link arm is directly coupled with the motor shaft. No speed reducer and or self-locking gear arrangement is used to provide holding torque to reduce the cost of laboratory setup implementation. Such arrangements are necessary for real industries. The hardware implementation block diagram is shown in Fig. 7.

A three-phase intelligent power module (IPM) is used to drive the motor, which is built with six IGBTs with isolation and gate driving unit and a diode rectifier with protection circuits and signal conditioning units. A single-phase autotransformer is used to supply the driving power from a 230 V AC line. The motor driving signals are conventionally generated by using programmable integrated chips. The coding is complex, costly, time consuming, and, therefore, difficult to handle. In recent years, researchers have used dSPACE-based DSP controllers to integrate a hardware–software platform for MATLAB/SIMULINK-based graphical programming. dSPACE DS1104 controller board is used with a connector panel to control real-time interfaces [13]–[16]. A host PC connected with the dSPACE is used to program through MATLAB/SIMULINK and monitor the real-time performance of the motor in terms of current, speed, and position through the CONTROL-DESK software virtual panel. Motor hall signals are fed to the controller via the hall sensor signal interface card to match voltage level. dSPACE DS1104 adaptor card is used to link the connector panel and IPM in properly specified connector standards.

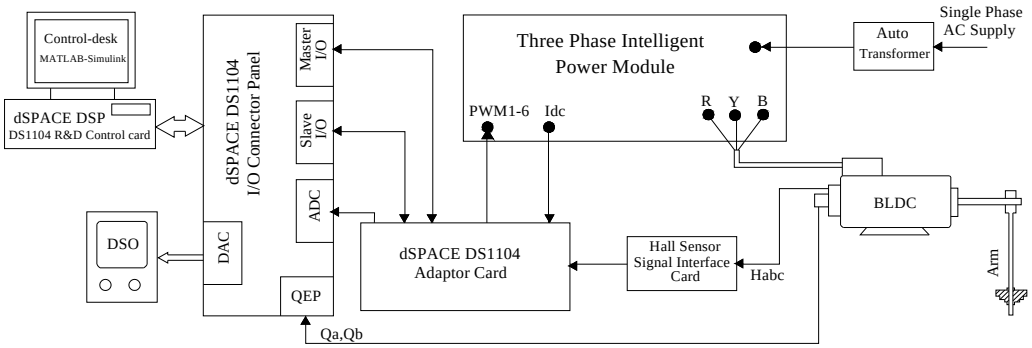


Fig. 7. dSPACE hardware implementation block diagram.

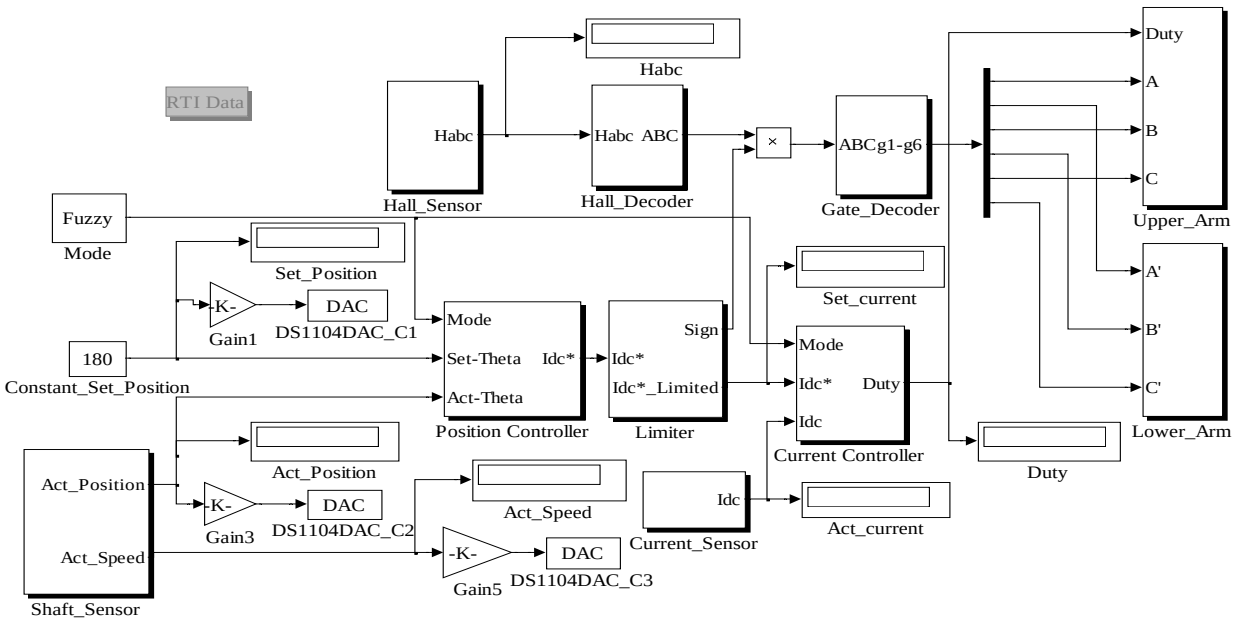


Fig. 8. MATLAB/SIMULINK based graphical programming.

Hall signals are acquired by the high-speed slave capturing unit, and gate PWM pulses are generated in slave DSP of dSPACE. Motor scaled current is acquired through 16-bit resolution ADC unit and motor coupled encoder signals are acquired through the QEP unit. DAC is used to generate position and speed equivalent voltage for monitoring through a digital storage oscilloscope.

SIMULINK-based graphical programming of the proposed drive is shown in Fig. 8. Hall signals decoded in the subsystem are used to generate six gate pulses in vendor-specified commutation table. In general, for 120° conduction mode of an inverter, two switches are maintained in the ON position at any instant—one from the upper arm and another from the lower arm—to ensure sufficient one-switch chopping to control the voltage to reduce the switching voltage stress. Upper arm switches are chopped at 10 KHz chopping frequency, that is, slave DSP and lower arm switches are kept ON and OFF through master digital outputs of dSPACE.

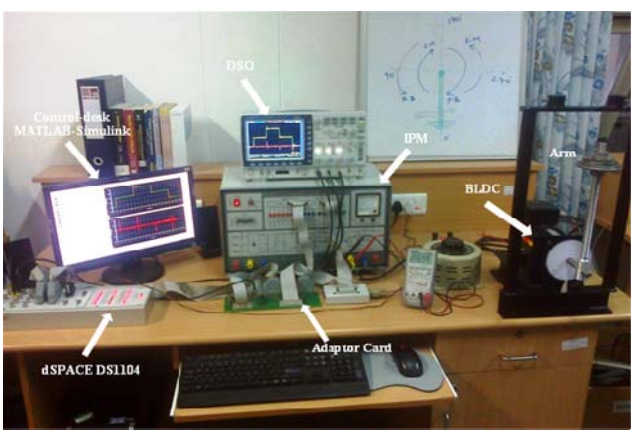


Fig. 9. Real-time implementation photo view.

The shaft sensor subsystem is used to measure the position and speed of the motor. The quadrature encoder model will increment or decrement 0.25 counts as per phase shift of quadrature signals Qa and Qb in each and every edge. The encoder has 2000 slots, and every slot gives one rising and

falling edge. Combining these two signals Qa and Qb ensures that four edges are possible per slot, which makes 4×0.25=1 pulse count. Hence, one revolution of the motor makes 2000 pulse counts and thus increases the resolution of the drive. Pulse counts are multiplied with a gain of 360/2000 to obtain the actual position in degrees. Speed can be calculated from the delta position, which provides the difference of the pulse count value from the last to the current sample step measured in the encoder lines and multiplied with a gain of 60/0.0001×2000, as shown in the following equation:

$$N = \frac{60}{T_s \times ppr} \times \Delta PulseCounts \tag{3}$$

Where N is the speed in rpm, T_s is the sampling time (0.0001 s), *ppr* is equal to 2000 pulse per revolution of the encoder, and Δ is the difference of the pulse count value between two samples.

The average value of 10 speed information samples is taken to avoid noise at low speed. The measured speed is used for monitoring only and not for any controlling purpose. The hall current sensor with the signal conditioning unit in IPM is used to measure the DC link current. The analog equivalent voltage of the DC link current is acquired through the ADC in a sample of 0.0001 s, as shown in the current sensor subsystem and is offset and scaled to obtain the original current magnitude. Fig. 9 shows the complete picture of hardware implementation.

V. Design of Controllers

The feedback controller is the heart of the servo driver. The design of the feedback controller requires two feedback closed loops: one is an inner loop current controller to limit the maximum DC link current to protect the motor, and the other one is a position controller in the outer loop to maintain the arm at the set/desired position. The conventional PID controller used in the feedback system is simple and easy to operate, but its performance is poor, unreliable, and highly influenced adversely by load disturbance and change in commands. Furthermore, the design of the control gain involves intricate mathematical modeling. By contrast, the fuzzy PID controller does not require complex mathematical modeling and has unlimited scope, which enables it to perform perfectly at a fast phase in all working environments without errors [17], [18].

The SIMULINK model of the conventional PID controller, as shown in Fig. 10, is used in both position and current controllers. Gain values K_p, K_i, and K_d differ in terms of position and current magnitude working ranges. According to the PID control law, the mathematical formulation of the discrete controller is obtained by Equ. (4). The control gain values are determined using Ziegler–Nichols tuning method. The tuned coefficients may be inadequate in all

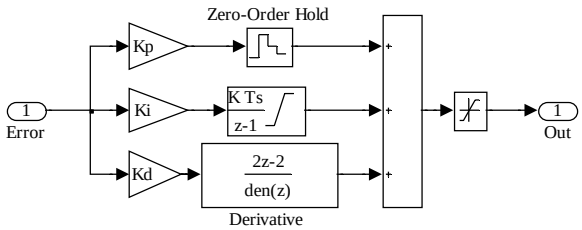


Fig. 10. Conventional PID controller.

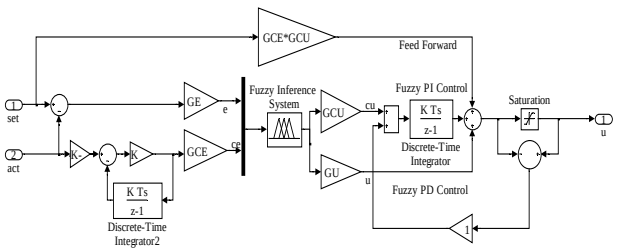


Fig. 11. Fuzzy PID controller.

environments; therefore, a gain-tuned controller and optimization techniques are essential in such situations. Instead of going through such a complex process, simply replacing conventional PID with fuzzy-based PID controllers [19]–[21] with the same gain equivalent scaling factors can improve the performance:

$$U_n = K_p E_n + K_i \sum_{i=1}^{n} E_i T_s + K_d \frac{E_n - E_{n-1}}{T_s} \tag{4}$$

Where K_p is the proportional gain, K_i is the integral gain, K_d is the derivative gain, and T_s is the sampling time in seconds.

Fig. 11 shows the combination of fuzzy PI with fuzzy PD to implement the fuzzy PID controller. An error and change of measurement are the inputs to the controller. Instead of using the change of error, change of measurement is utilized to prevent the step change in reference signal from directly triggering the derivative action. The range of membership function used is unity ±1; hence, scaling the control signals is necessary before and after the fuzzy inference. Reference current magnitude ±Idc* is the output of the outer loop position controller, and gate pulse duty cycle (0 to 1) is the output of the inner loop current controller.

Inputs are scaled through error normalization factor GE and change in measurement normalization factor GCE. The output response signal is scaled through response de-normalization factor GU and change in response de-normalization factor GCU. These scaling factors play a vital role in designing the fuzzy PID controller. A feed-forward path is provided to ensure proportional action when the fuzzy PID controller is linear. An anti-windup feature is included to control the saturation level of the output response signal to ensure the quick response of the fuzzy PID controller. The scaling factors of the fuzzy PID can be derived from the conventional PID gain values through a linear approximation method [22]. The response of fuzzy PI

controller is obtained by leaving out the feed-forward and anti-windup features, and considering only the fuzzy PI control path, as shown in Equ. (5):

$$U_n(F_{PI}) = GCU * \sum_{i=1}^{n}\left[GE * E_i + GCE * \frac{E_i - E_{i-1}}{T_s}\right]T_s$$
$$= GCU * \left[GE * \sum_{i=1}^{n} E_i T_s + GCE * \sum_{i=1}^{n}(E_i - E_{i-1})\right] \quad (5)$$
$$= GCU * GE * \sum_{i=1}^{n} E_i T_s + GCU * GCE * E_n$$

Where *GE* is the error normalization factor, *GCE* is the change in measurement normalization factor, *GU* is the response to the de-normalization factor, and *GCU* is the change in response to the de-normalization factor.

Similarly, with only the fuzzy PD path considered, the response can be written as

$$U_n(F_{PD}) = GU * \left[GE * E_n + GCE * \frac{E_n - E_{n-1}}{T_s}\right]$$
$$= GU * GE * E_n + GU * GCE * \frac{E_n - E_{n-1}}{T_s}. \quad (6)$$

Adding Equs. (5) and (6) to form fuzzy PID will give

$$U_n(F_{PID}) = U_n(F_{PI}) + U_n(F_{PD})$$
$$U_n(F_{PID}) = (GCU * GCE + GU * GE) * E_n$$
$$+ (GCU * GE) * \sum_{i=1}^{n} E_i T_s + (GU * GCE) * \frac{E_n - E_{n-1}}{T_s}. \quad (7)$$

Through a comparison between Equs. (7) and (4), the relations between conventional PID gains with fuzzy scaling factors are formed

$$K_p = GCU * GCE + GU * GE \quad (8)$$
$$K_i = GCU * GE \quad (9)$$
$$K_d = GU * GCE \quad (10)$$

Error normalization factor GE is fixed with the reciprocal of the maximum input error as per Equ. (11), and the other scaling factors are shown in Equs. (12)–(14), which are computed from Equs. (8)–(10)

$$GE = \frac{1}{\max .error} \quad (11)$$
$$GCE = GE * \left(K_p - \sqrt{K_p^2 - 4K_i K_d}\right)\frac{K_i}{2} \quad (12)$$
$$GCU = \frac{K_i}{GE} \quad (13)$$
$$GU = \frac{K_d}{GCE} \quad (14)$$

Input and output variables are mapped via membership functions to work with fuzzy inference system. These variables are simple triangular and cross-neighbor sets with a membership value of 0.5, as shown in Fig. 12. Mamdani-type inference is used for inference engine, and center of gravity method is used for defuzzification.

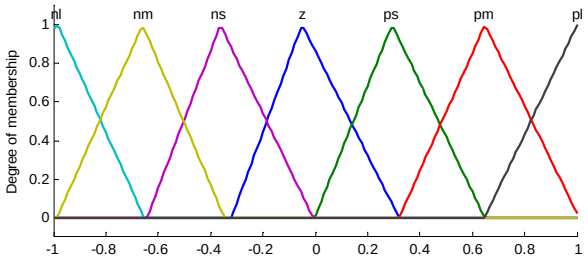


Fig. 12. Membership functions of 'e', 'ce' and 'u'.

TABLE II
FUZZY RULE-BASED MATRIX

ce \ e	nl	nm	ns	z	ps	pm	pl
nl	nl	nl	nl	nl	nm	ns	z
nm	nl	nl	nl	nm	ns	z	ps
ns	nl	nl	nm	ns	z	ps	pm
z	nl	nm	ns	z	ps	pm	pl
ps	nm	ns	z	ps	pm	pl	pl
pm	ns	z	ps	pm	pl	pl	pl
pl	z	ps	pm	pl	pl	pl	pl

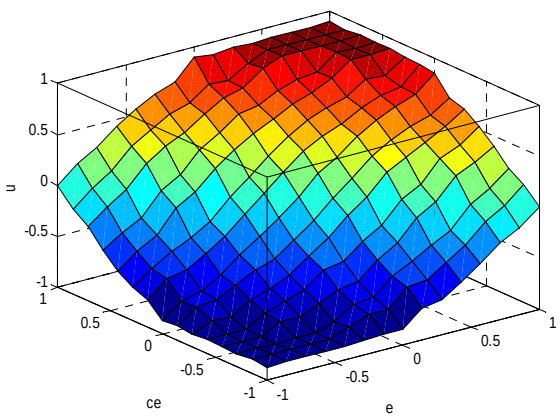


Fig. 13. Surface view.

The linguistic variables are divided into seven groups, namely, negative large (nl), negative medium (nm), negative small (ns), zero (z), positive small (ps), positive medium (pm), and positive large (pl) [23].

The work of inference engine is based on the 7×7=49 rules, as shown in the matrix in Table II. The top row and left column of the matrix indicate the fuzzy set of the variables "e" and "ce," respectively, and variable "u" is shown in the body of the matrix. The surface view of the fuzzy PID controller is shown in Fig. 13, where the x-axis is error "e," the y-axis is the change in measurement "ce," and the z-axis is output response "u." The resultant plot shows the smooth surface.

The same design used both for conventional PID and fuzzy PID-based feedback controllers discussed above in detail are utilized for simulation and hardware implementation.

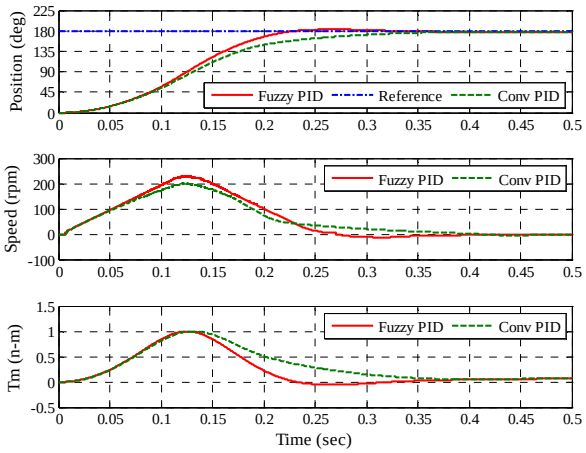


Fig. 14. Forward motoring mode (0°–180°).

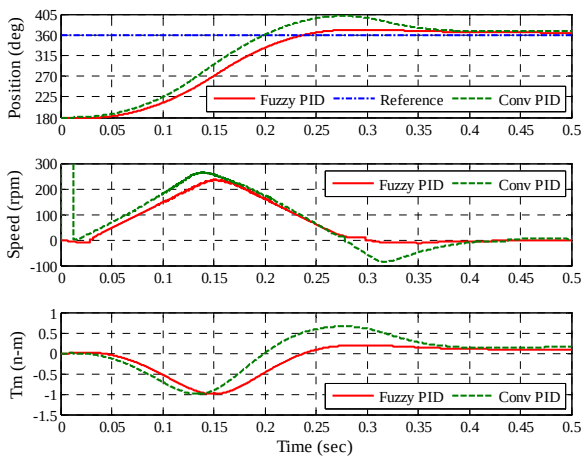


Fig. 15. Forward braking mode (180°–360°).

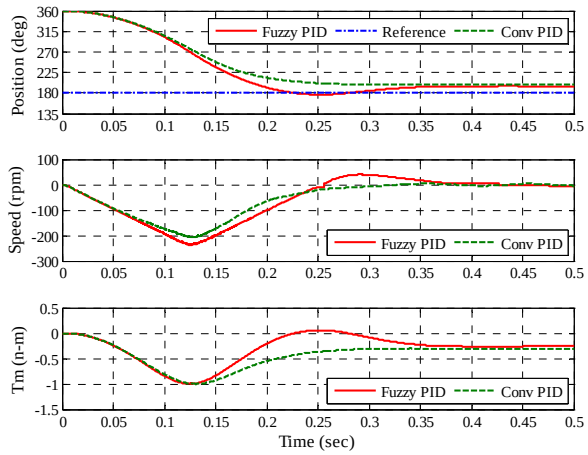


Fig. 16. Reverse motoring mode (360°–180°).

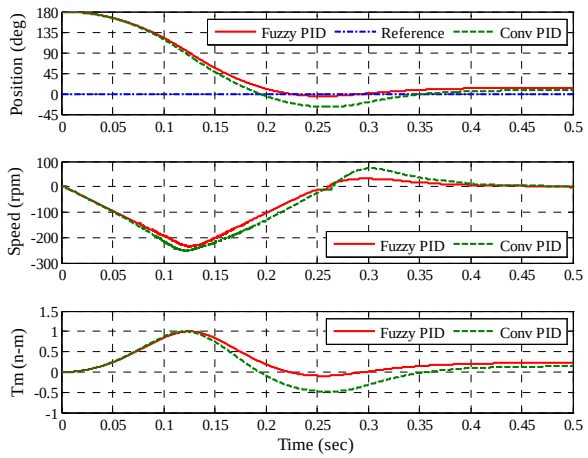


Fig. 17. Reverse braking mode (180°–0°).

VI. Results and Discussions

The single-arm position servo drive is simulated, and the hardware is implemented in all four quadrants. Several tests are performed to verify the dynamic behavior of both conventional PID and fuzzy PID controllers.

A. Simulation Results

Simulation is performed up to 0.5 s, and the initial position is altered in the requisite quadrant for the sake of the analysis. A payload torque of 1 n–m is applied to the arm in all the four quadrants.

When the arm moves upward in a clockwise direction from 0° to 180°, as shown in Fig. 14, drive operation takes place in the first quadrant of FM. The speed and torque are the same in positive sign. The torque would be higher at 90° and the speed starts reducing near 90°. The conventional PID needs 0.35 s of settling time (Ts), whereas the fuzzy PID takes only 0.25 s. A cursory comparison of the dynamic behaviors listed in Table III rising time (Tr), percentage of maximum overshoot (Mp), and ± percentage of the steady-state error clearly shows that the fuzzy PID controller is a better version than the conventional PID controller.

During the downward movements of the arm in the clockwise direction (180°–360°), as shown in Fig. 15, the speed and torque are in the opposite sign to each other, driving the motor in the second quadrant of FB. The payload torque pulls down the arm faster than the safe limit speed. The settling time of FB with conventional PID is 0.2 s, which is faster than in the FM mode because of the effect of gravitational force. The same is 0.25 s with fuzzy PID, which is equal to the time taken in FM mode. The dynamic behavior of fuzzy PID also reveals that fuzzy PID is safer and a robust controller in both quadrants.

In the third quadrant of the RM mode, as shown in Fig. 16, the arm travels from 360° to 180° (0° to -180°) upward in the anticlockwise direction. The speed and torque signs are negative. The arm needs more time to settle at the reference position of 180° because of increasing torque with respect to position. While the conventional PID needs 0.35 s of settling time, the fuzzy PID settles in 0.25 s, which is the same time taken in the FM mode.

When the arm rotates downward from 180° to 0° (-180° – -360°) in the anticlockwise direction, as shown in Fig. 17, the payload torque pulls down the arm faster than the safer speed

TABLE III
SIMULATION RESULTS COMPARISON

Controller	Conventional PID				Fuzzy PID			
Parameter / Set Position	Tr (s)	Mp (%)	Ts (s)	± Error (%)	Tr (s)	Mp (%)	Ts (s)	± Error (%)
0° to 180° (Forward Motoring)	0.2	0	0.35	-6.66	0.15	1.11	0.25	+2.22
180° to 360° (Forward Braking)	0.1	22.22	0.2	+5.55	0.15	5.55	0.25	+2.66
360° to 180° (Reverse Motoring)	0.2	0	0.35	-5.55	0.15	1.11	0.25	-3.33
180° to 0° (Reverse Braking)	0.1	22.22	0.2	-3.33	0.15	5.55	0.25	-4.44

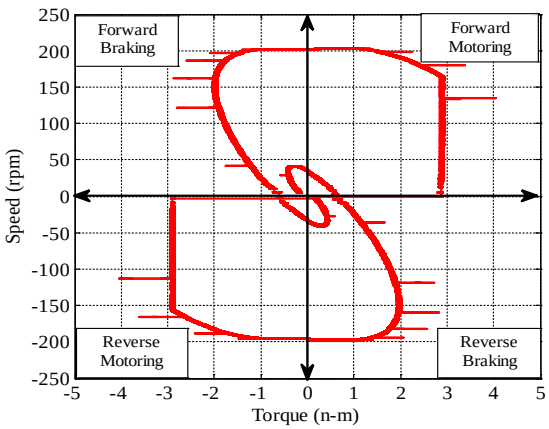


Fig. 18. Speed–torque characteristics.

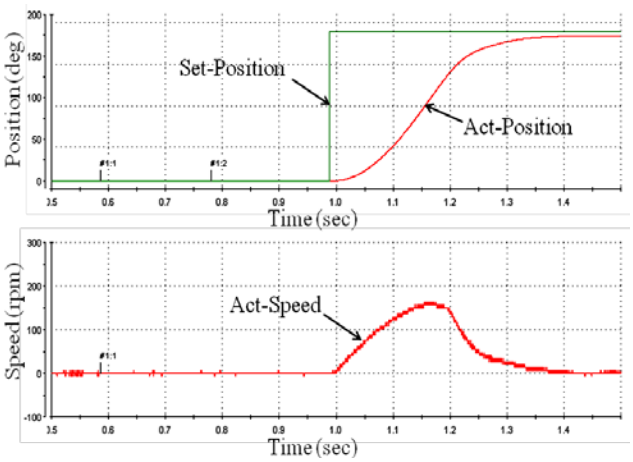
limit, thereby operating the drive in the fourth quadrant of RB mode. If the arm travels faster in the downward direction in 0.2 s with the conventional PID controller, then the arm is likely to drop the payload and/or crash to the floor, thereby damaging the arm and/or the payload. However, the descent is reached in 0.25 s with the fuzzy PID by taking the same time taken in the motoring mode.

Thus, the comparative study of the dynamic behavior of the controllers in all the four quadrants, as shown in Table III, amply proves that the fuzzy PID controller is not only robust but also provide a better performance than the conventional PID controller. Speed versus torque characteristics of the vertically rotating single-arm driving BLDC motor in all the four quadrants are shown in Fig. 18.

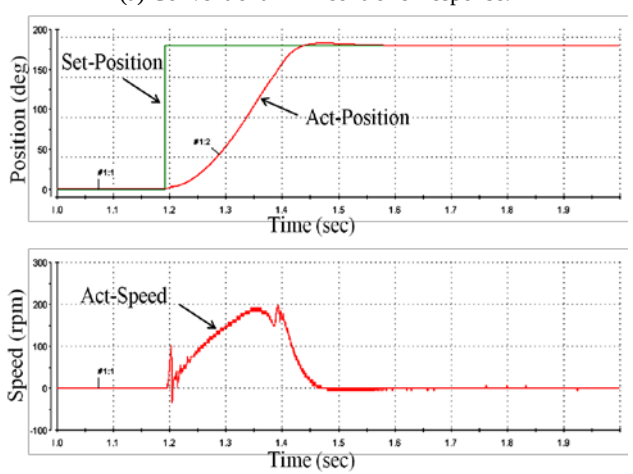
B. Experimental Results

Several tests were conducted with both the conventional PID and fuzzy PID controllers in the laboratory with a 35 cm-long single-arm that carries a constant 1 n–m payload torque equivalent weight in all four quadrants. The set position and actual position of the arm and the corresponding speed were captured in CONTROL-DESK.

When the arm moves upward in the clockwise direction from 0° to 180°, the motor rotates in the forward direction. To enable the arm to reach 90°, maximum torque is required



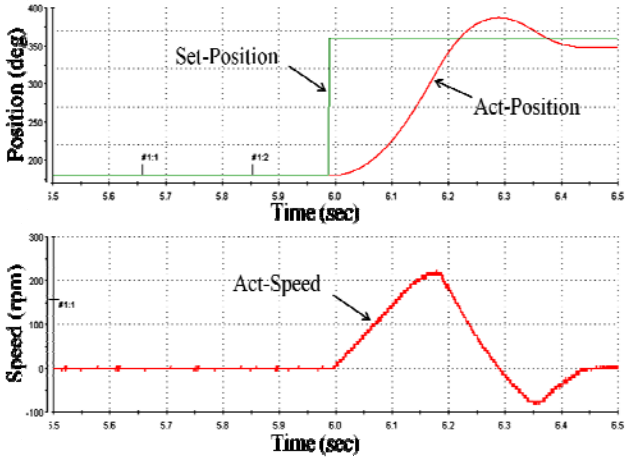
(a) Conventional PID controller response.



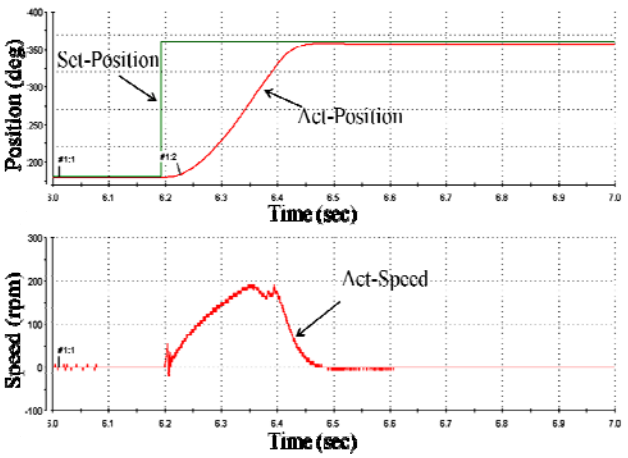
(b) Fuzzy PID controller response.

Fig. 19. Forward motoring mode (0°–180°).

because of the gravitational force, which causes the arm to pull down the payload to reach 180° at a reduced speed. Therefore, reaching the desired position takes a longer time. The inner-loop current controller limits the maximum safe current when increasing the payload torque. The outer loop position controller attempts to achieve the desired 180° with some delay. The conventional PID controller takes 0.37 s to reach the desired position, as shown in Fig. 19(a). The fuzzy PID

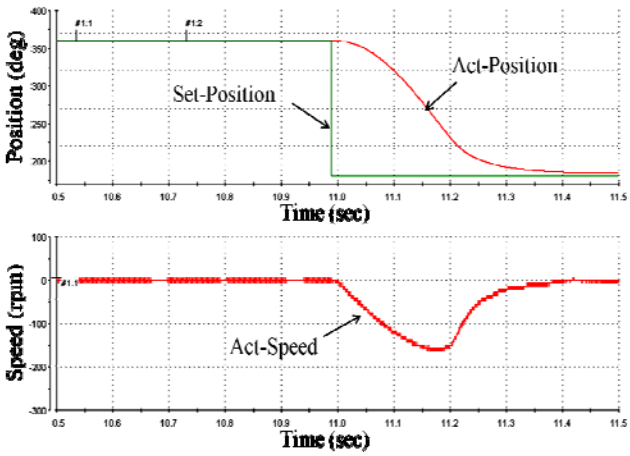


(a) Conventional PID controller response.

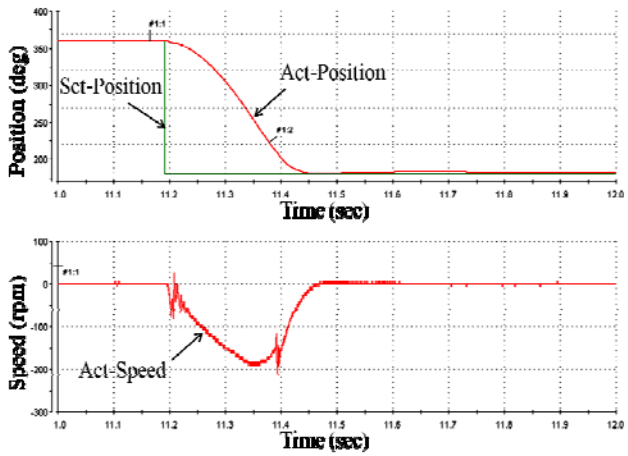


(b) Fuzzy PID controller response.

Fig. 20. Forward braking mode (180°–360°).



(a) Conventional PID controller response.



(b) Fuzzy PID controller response.

Fig. 21. Reverse motoring mode (360° to 180°).

controller takes 0.25 s settling time (Ts), as shown in Fig. 19(b). The comparative study results that relate to the dynamic behavior, such as rising time (Tr), maximum overshoot (Mp), and ± percentage of the steady-state error listed in Table IV, proves that the fuzzy PID controller provides a better performance than the conventional PID controller.

When the arm moves downward from 180° to 360° in the clockwise direction, it moves faster than that in the upward motion because of gravitational force. It operates in the second quadrant of FB at a safer speed to avoid accidents. With the conventional PID, the arm takes 0.21 s, as shown in Fig. 20(a) which is faster than the time taken in the motoring mode. However, the fuzzy PID controller takes 0.25 s, as shown in Fig. 20(b), which is the same time taken in motoring mode.

When the arm rotates in the reverse direction, i.e., 360°–180° upward anticlockwise, the motor must operate in the third quadrant of the RM mode. The time taken by the conventional PID is 0.37 s. as shown in Fig. 21(a), whereas the fuzzy PID controller takes 0.25 s, as shown in Fig. 21(b), which is same time taken in the FM mode.

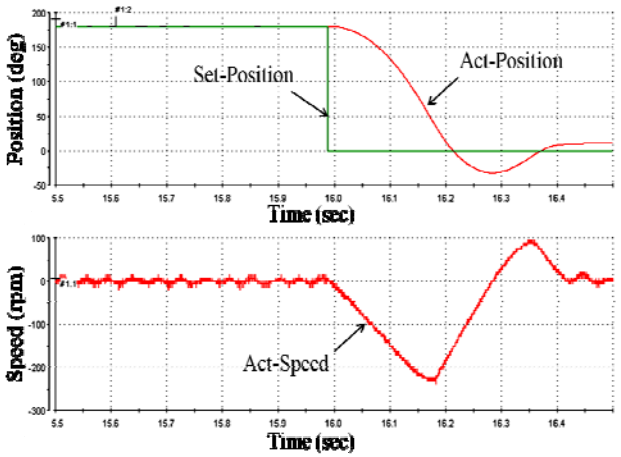
When the arm moves downward in the anticlockwise direction of 180°–0° in the fourth quadrant of the RB mode, the settling time taken is the same as in FB mode with both controllers, as shown in Figs. 22(a) and 22(b). During the braking mode of the operation with conventional PID, the controller produces maximum overshoot around 13.88% and more oscillations. The results of the comparative study in all four quadrants, as presented in Table IV, show that the fuzzy PID controller has fewer steady-state errors and takes the uniform settling time in all four quadrants.

Load impact analysis is conducted with steady-state condition. As revealed by the analysis, when the arm is at 90°, needed maximum torque with 1 n–m connected load. Furthermore, 1 n–m load torque equivalent weight is added and removed after 4 s. The sudden addition and removal of the load cause the arm to pull down and push up, respectively, with some percentage of position error.

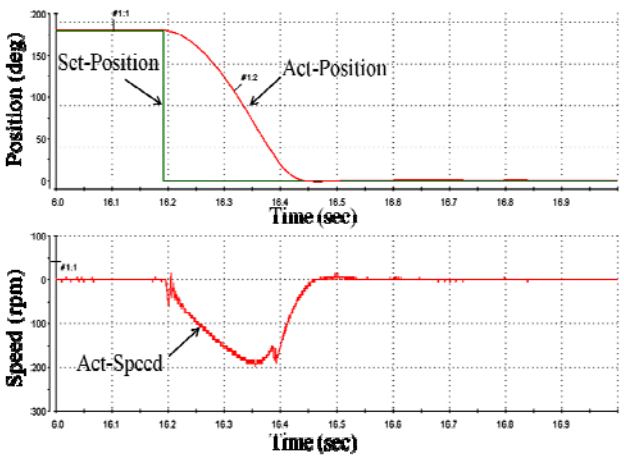
The alteration of the load to the arm with conventional PID controller affects the position up to ±25°, and the arm reaches a steady state after 1 s, with an error of 3%, as shown in Fig. 23(a). Reaching the steady state at 0.5 s without any error by

TABLE IV
EXPERIMENTAL RESULTS COMPARISON

Controller	Conventional PID				Fuzzy PID			
Parameter / Set Position	Tr (s)	Mp (%)	Ts (s)	± Error (%)	Tr (s)	Mp (%)	Ts (s)	± Error (%)
0° to 180° (Forward Motoring)	0.25	0	0.37	-3.77	0.15	2.77	0.25	-0.22
180° to 360° (Forward Braking)	0.15	13.88	0.21	-5.55	0.15	0	0.25	-3.55
360° to 180° (Reverse Motoring)	0.25	0	0.37	-3.77	0.15	0	0.25	-3.55
180° to 0° (Reverse Braking)	0.15	13.88	0.21	-5.55	0.15	1.77	0.25	-2.44

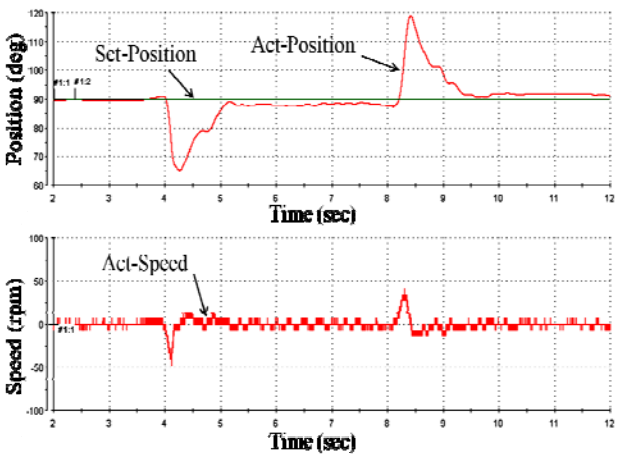


(a) Conventional PID controller response.

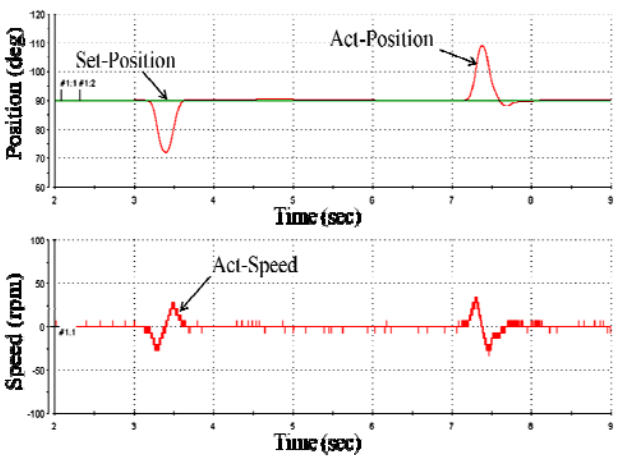


(b) Fuzzy PID controller response.

Fig. 22. Reverse braking mode (180°–0°).



(a) Response of the conventional PID controller.



(b) Response of the fuzzy PID controller.

Fig. 23. Analysis of the influence of the load.

changing the position by ±15° is the effect of the load alteration to the arm using the fuzzy PID controller, as shown in Fig. 23(b). Thus, the analysis of load impact proves that fuzzy PID controller is a robust and a better controller than the conventional PID controller. The practical implementation presents a small position error because of several parameters of the equipments.

The error is reduced significantly in the fuzzy PID controller. The comparative study of simulation results, as shown in Table III, with the experimental results, as shown in Table IV, shows that dSPACE DS1104-based hardware implementation results are close to the simulation results. Hence implemented hardware is validated.

VII. CONCLUSION

In this paper, a control scheme has been proposed for a position servo drive for a vertically rotating single arm of a robot with varying load torque conditions with operations in all four quadrants. Simulation and experimental results show that the fuzzy PID controller is a dependable and robust controller, which ensures that the same settling time is maintained by the arm in all four quadrants. In other words, the fuzzy PID controller is proven to be the most suitable feedback controller for controlling the position of the vertical rotating robotic arm in modern servo drives.

REFERENCES

[1] T. J. E. Miller, *Brushless Permanent Magnet and Reluctance Motor Drives*, Oxford Science Publisher, 1989.

[2] T. Kenjo and S. Nagamori, *Permanent Magnet Brushless DC Motors*, Oxford Science Publisher, 1985.

[3] B. K. Bose, *Power Electronics and Variable Frequency Drives*, Wiley-Blackwell, 1996.

[4] R. J. Wai and M. C. Lee, "Intelligent optimal control of single-link flexible robot arm," *IEEE Trans. Indu. Electron*, Vol. 51, No. 1, pp. 201-220, Feb. 2004.

[5] M. T. Soylemez, M. Gokasan, and O. S. Bogosyan, "Position control of a single-link robot-arm using a multi loop PI controller," in *IEEE Conference on Control Applications*, Vol. 2, pp. 1001-1006, 2003.

[6] A. Manoharaprasad and C. V. N. Raja, "Control of two-link scara robot using IMC PID and fuzzy controllers," *International Journal of Emerging Trends in Engineering and Development*, Vol. 2, No. 1, pp. 248-258, Mar. 2014.

[7] A. Z. Alassar, I. M. Abuhadrous, and H. A. Elaydi, "Comparison between FLC and PID controller for 5 DOF robot arm," in *International Conference on Advanced Computer Control*, Vol. 5, pp. 277-281, 2010.

[8] R. D. Klafter, A. Thomas, A. Chmielewski, and M. Negin, *Robotic Engineering an Integrated Approach*, PHI Learning, Chap. 2, pp. 87-97, 2009.

[9] U. Vinatha and P. Swetha, "Simulation of four quadrant operation & speed control of BLDC motor on MATLAB/SIMULINK," in *IEEE Conference on TENCON*, pp. 1-6, 2008.

[10] C. S. Joice, S. R. Paranjothi, and V. J. S. Kumar, "Digital control strategy for four quadrant operation of three phase BLDC motor with load variations," *IEEE Trans. Ind. Informat*, Vol. 9, No. 2, pp. 974-982, May 2013.

[11] V. M. Hernandez-Guzman, V. Santibanez, and R. Campa, "PID control of rigid robots actuated by Brushless DC motors," in *American Control Conference*, pp. 1430-1435, 2008.

[12] R. Manikandan and R. Arulmozhiyal, "Modeling and implementation of quadrature encoder for servo drive applications," *International Journal of Applied Engineering Research*, Vol. 10, No. 10, pp. 9711-9717, Apr. 2015.

[13] R. Fernando and E. Ali, "A novel digital control technique for Brushless DC motor drives," *IEEE Trans. Indus Electron*, Vol. 54, No. 5, pp. 2365-2373, Oct. 2007.

[14] B. Subudhi, A. K. Anish Kumar, and D. Jena, "DSPACE implementation of fuzzy logic based vector control of induction motor," in *IEEE Region 10 Conference TENCON*, pp.1-6, 2008.

[15] A. Rubaai, M. J. Castro-Sitiriche, and A. R. Ofoli, "DSP-based laboratory implementation of hybrid fuzzy-PID controller using genetic optimization for high-performance motor drives," *IEEE Trans. Ind. Appl*, Vol. 44, No. 6, pp. 1977-1986, Nov. 2008.

[16] M. Hu, J. Qiu, and C. Shi, "A comparative analysis of fuzzy PI and PI speed control in Brushless DC motor based on dSPACE," in *International Conference on Electrical Machines and Systems*, pp. 1-5, 2011.

[17] L. A. Zadeh, "Fuzzy sets," *Information and Control*, Vol. 8, No. 3, pp. 338-353, Jun. 1965.

[18] X. J. Xu, C. C. Hang, and C. Liu, "Parallel structure and tuning of a fuzzy PID controller," *Automatica*, Vol. 36, No. 5, pp. 673-684, May 2000.

[19] L. V. Hongli, D. Peiyong, C. Wenjian, and J. Lei, "Direct conversion of PID controller to fuzzy controller method for robustness," in *IEEE Conference on Industrial Electronics and Applications*, pp.790-794, 2008.

[20] R. Arulmozhiyal and K. Baskaran, "Implementation of a fuzzy PI controller for speed control of induction motors using FPGA," *Journal of Power Electronics*, Vol. 10. No. 1, pp. 65-71, Jan. 2010.

[21] A. G. Sreenatha and P. Makarand, "Fuzzy logic controller for position control of flexible structures," *Acta Astronaut Journal*, Vol. 50, No. 11, pp. 665-671, Jun. 2002.

[22] J. Jantzen, "Tuning of Fuzzy PID Controllers," Technical Report, University of Denmark, pp. 1-22, Apr. 1999.

[23] R. Manikandan and R. Arulmozhiyal, "Position control of DC servo drive using fuzzy logic controller," in *IEEE International Conference on Advances in Electrical Engineering (ICAEE)*, pp.1-5, 2014.

Manikandan R. was born in India in 1986. He received his B.E. degree in Electrical and Electronics Engineering from Mahendra Engineering College, Namakkal, India, in 2008 and his M.E. degree in Power Electronics and Drives from Sona College of Technology, Salem, India, in 2010. He is currently a Research Associate at SONAPEDAC R&D Centre, Department of Electrical and Electronics Engineering, Sona College of Technology. He is currently working toward his Ph.D. degree in Electrical Engineering at Sona College of Technology, under Anna University, Chennai, India. His current research interests include special machine drives, fuzzy-based controllers, dSPACE, and FPGA implementation. He is a life member of ISTE.

Arulmozhiyal R. was born in India in 1973. She received her B.E. and M.E. degrees in Electrical and Electronics Engineering from the University of the Madras, India, in 1999 and 2006, respectively, and her Ph.D. degree in Electrical Engineering from Anna University of Technology, Coimbatore, India, in 2011. She is currently a Professor in the Department of Electrical and Electronics Engineering, Sona College of Technology, Salem, India. Prof. R. Arulmozhiyal is the Head of SONAPEDAC R&D Centre, Department of Electrical and Electronics Engineering, Sona College of Technology. Her research interests include AI techniques for solid-state drives and fin control actuation systems. She is a senior member of the IEEE, a life member of the ISTE and a fellow member of the IE (India).