Design and Implementation of Parallel Fuzzy PID Controller for High-Performance Brushless Motor Drives: An Integrated Environment for Rapid Control Prototyping

Ahmed Rubaai, *Senior Member, IEEE*, Marcel J. Castro-Sitiriche, *Student Member, IEEE*, and Abdul R. Ofoli, *Member, IEEE*

***Abstract*—This paper presents an integrated environment for the rapid prototyping of a robust fuzzy proportional–integral–derivative (PID) controller that allows rapid realization of novel designs. Both the design of the fuzzy PID controller and its integration with the classical PID in a global control system are developed. The architecture of the fuzzy PID controller is basically composed of three parallel fuzzy subcontrollers. Then, the parallel subcontrollers are grouped together to form the overall fuzzy PID controller. The fuzzy proportional, integral, and derivative gains are direct output from the parallel fuzzy subcontrollers and are derived in the error domain. Thus, the proposed architecture presents an alternative to control schemes employed so far. The integrated controller is formulated and implemented in real time, using the speed control of a brushless drive system as a test bed. The design, analysis, and implementation stages are carried out entirely using a dSPACE DS1104 digital-signal-processor-based real-time data acquisition control system and MATLAB/Simulink environment. Experimental results show that the proposed hybrid fuzzy PID controller produces superior control performance than the conventional PID controllers, particularly in handling nonlinearities and external disturbances.**

***Index Terms*—Classical proportional–integral–derivative (PID) control, dSPACE digital signal processor (DSP), hybrid fuzzy PID controller, MATLAB/Simulink, robustness.**

I. Introduction

WITH THE advent of digital signal processors (DSPs) and high-performance processors, digital technology has greatly impacted the motion control industry. The availability of high-level languages has allowed the introduction of software design methodologies and testing procedures that have both boosted the reliability of designs and shortened the design cycle. Classical proportional–integral–derivative (PID) controllers are commonly used in industry due to their simplicity, clear functionality, and ease of implementation [1]–[5]. New results on the synthesis of PID controllers have been reported in [1], i.e., validates the call for innovative studies in the area of PID controllers. A multiresolution PID controller-based wavelet transform has been developed and tested for two different motion control problems [2]. A model free gain-tuning method to enhance the performance of classical PI controller has been proposed and implemented for a permanent magnet synchronized motor drive system [3]. The method is based on the cycle information. Clearly, more research efforts are needed to realize the future of both methods [2], [3]. Furthermore, it has been reported in [4] that more than 96% of control loops employ conventional PID controllers. However, many practitioners switch off or even exclude the derivative term by setting the derivative gain to zero [5]. A collection of five papers on PID control was published in the February 2006 issue of the *IEEE Control Systems Magazine* [6]. The papers examine PID control as it is practiced today and explore recent innovations that point toward the future. The process for designing PID controllers for industrial automation is quite elaborated and can be difficult in practice, if multiple and conflicting objectives are to be achieved [4]. This stimulates the development of some advanced tuning methods that can be incorporated in hardware modules, and the search is on to find the next key technology or methodology for PID tuning [7].

Fuzzy set theory plays an important role in dealing with uncertainty when making decisions in complex or ill-defined processes [8]–[13]. The concept of optimal fuzzy reasoning is introduced in [8]. A fuzzy-based two-loop model-following control system has been presented for robust PID controllers [9]. The method is tested on the temperature control of a multizonal laboratory electroheated process. A function-based evaluation technique has been illustrated for a systematic study of PID-like controllers [10]. A flexible complexity reduced PID-like fuzzy controller in which the functional scaling factors are heuristically generated has been introduced [11]. However, Xu *et al.* [12] discusses the feasibility of parallel structure and tuning of a fuzzy PID controller. It combines PI and PD to form the fuzzy PID controller. The control signal is directly deduced from the fuzzy inference. However, only analytical

Paper MSDAD-07-61, presented at the 2006 Industry Applications Society Annual Meeting, Tampa, FL, October 8–12, and approved for publication in the IEEE TRANSACTIONS ON INDUSTRY APPLICATIONS by the Industrial Automation and Control Committee of the IEEE Industry Applications Society. Manuscript submitted for review October 31, 2006 and released for publication October 26, 2007. Published July 23, 2008 (projected).

A. Rubaai is with the Electrical and Computer Engineering Department and the Multipurpose and Multidisciplinary Control System Laboratory, Howard University, Washington, DC 20059 USA (e-mail: arubaai@howard.edu).

M. J. Castro-Sitiriche is with the Electrical and Computer Engineering Department, Howard University, Washington, DC 20059 USA.

A. R. Ofoli is with Cummins Inc., Columbus, IN 47202 USA.

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TIA.2008.926059

and theoretical development has been presented. The paper also lacks experimental and practical implementation. An attempt to undertake the development of a new methodology for the design of fuzzy PID controllers has been described in [13]. The methodology is based on theoretical fuzzy analysis and genetic-based optimization. Fuzzy control has recently found extensive application for motion control industry and has attracted the growing attention and interest of many control researchers due to model free concept [14], [15]. The results suggest that fuzzy logic control is highly suitable for motion controls and drives. Most controller manufacturers have incorporated their knowledge base that is applied to the development of their own products into their algorithms.

Due to the lack of a widely applicable tuning method, a need for the development of comprehensive and easy to use tuning controllers has therefore arisen. Although this is a critical issue in motion control industry, very few papers address the problem. As a result, a complete and integrated environment is required to support a designer throughout the development of a control system, from initial design phase until the final steps of code generation. In response, this paper proposes an integrated environment for the rapid control prototyping of a robust fuzzy PID controller using a state-of-the-art dSPACE DS1104 DSP-based data acquisition control (DAC) system and MATLAB/Simulink environment. Unlike previous works, the fuzzy PID controller is composed of three parallel fuzzy subcontrollers. Consequently, the three fuzzy subcontrollers are combined together to form the overall fuzzy PID controller. The fuzzy proportional, integral, and derivative gains are direct output from the parallel fuzzy subcontrollers and are derived in the error domain. Hence, the gains are directly realized from the knowledge base and the fuzzy inference, and then, each subcontroller generates the control signal. The delayed control signal is then used as a feedback signal and employed to tune online the gains in terms of knowledge base and fuzzy inference. The tuning mechanism for the fuzzy proportional, integral, and derivative gains is executed such that the overall control signal is less than the saturation limit at all times, thus reducing or even eliminating the core source of windup if any. Experimental results have shown that the proposed fuzzy PID controller offers better performance than the conventional PID control.

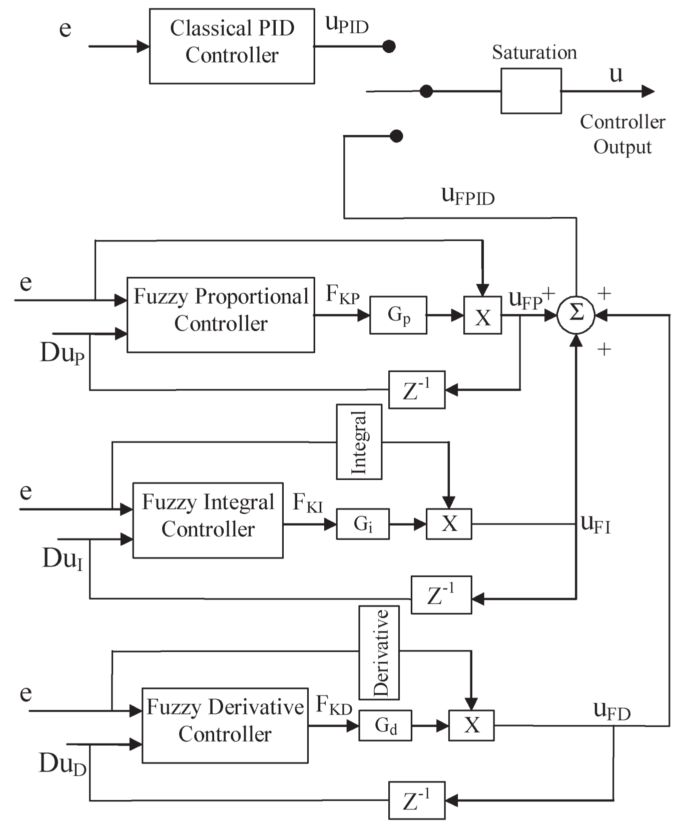


Fig. 1. Implementation of the hybrid fuzzy PID controller.

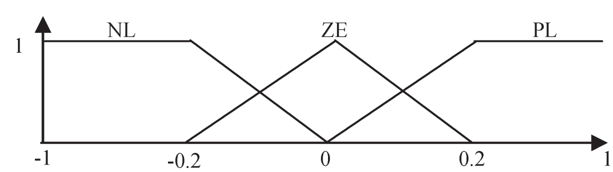


Fig. 2. Membership functions for speed error.

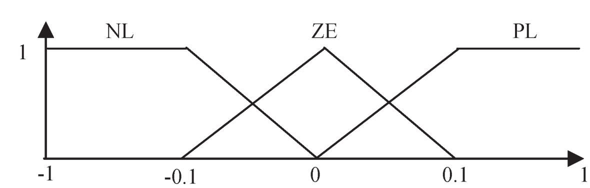


Fig. 3. Membership functions for the delayed control signals.

II. Proposed Fuzzy PID Controller

The proposed fuzzy PID controller consists of three parallel fuzzy subcontrollers that update online the values of the proportional, integral, and derivative gains. Fig. 1 shows the architecture of the proposed fuzzy PID controller and its integration with the classical PID. Each of the fuzzy subcontrollers has two inputs and one output. The inputs are the rotor speed error e and the delayed feedback control signal Du with each of these inputs corresponding to a fuzzy variable. Note that the delayed control signal Du consists of the three input signals Du_P, Du_I, and Du_D, each corresponds to a fuzzy subcontroller. The fuzzy variable associated with the rotor speed error consists of three fuzzy sets: positive large (PL), zero (ZE), and negative large (NL). In this way, for any given value of the rotor speed error, the degree of membership μ, to which it belongs to each of these sets, can be determined, and a control decision based on this information can be obtained. Similarly, each delayed feedback control signal can be described by a group of partially overlapping fuzzy sets as PL, ZE, and NL, with each set having its own membership function. Typical triangular membership functions are utilized for the speed error and each of the delayed feedback control signals. Figs. 2 and 3 show the membership functions for the fuzzy inputs. The three fuzzy sets can be symbolized by F_i^j, $i = 1, 2$ and $j = 1, 2, 3$. Their corresponding membership functions can be symbolized by $\mu F^j(e, \mathrm{Du})$, $j = 1, 2, 3$. The output of the fuzzy control decision that corresponds to each fuzzy subcontroller is the fuzzy proportional, integral, and derivative gains (FK_P, FK_I, and FK_D). Clearly, FK_P, FK_I, and FK_D are direct output from

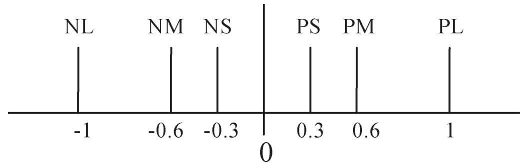


Fig. 4. Membership functions of the fuzzy output.

the parallel fuzzy subcontrollers and are directly realized from the knowledge base and the fuzzy inference. The membership functions for the output fuzzy sets of each subcontroller have singleton shapes. Fig. 4 shows the membership functions for the fuzzy output. The singleton output membership functions are used because of the efficient combination with the weighted average defuzzification process.

The next step in the design of the fuzzy subcontroller is the determination of the fuzzy IF–THEN inference rules. The number of fuzzy rules that are required is equal to the product of the number of fuzzy sets that make up each of the two fuzzy input variables. For each fuzzy subcontroller described here, the input variables representing the rotor speed error and the corresponding delayed control signal consists of three fuzzy sets each. Thus, a total of nine fuzzy rules are required. The fuzzy inference rules have been determined experimentally using the measured responses of the brushless servo drive system. A typical fuzzy rule is of the following form: $R^{(1)}$: IF e is F_1^j, and Du is F_2^j THEN f is C_1^l, for $j = 1, \ldots, 3, l = 1, \ldots, 3$.

In general: $R^{(k)}$: IF e is F_1^j, and Du is F_2^j THEN f is C_k^l for $j = 1, \ldots, 3$, $l = 1, \ldots, 3$, $k = 1, \ldots, 9$.

The conjunction of the rule antecedents is evaluated by the fuzzy operation *intersection* which is implemented by the min operator. The rule strength represents the degree of membership of the output variable for a particular rule. The rule strength $\xi_{i,j}$ of a particular rule is defined as

$$\xi_{i,j} = \min(\mu_{F_i^e}, \mu_{F_j^{\mathrm{Du}}}) \tag{1}$$

where $i \in [\mathrm{PB}, \mathrm{ZE}, \mathrm{NB}]$ is associated with the fuzzy variable, speed error, and $j \in [\mathrm{PB}, \mathrm{ZE}, \mathrm{NB}]$ is associated with the fuzzy variable, delayed feedback control. From the aforementioned, it is clear that for all the rules where at least one of the degrees of membership in the corresponding fuzzy sets are zero, the output of the min operator will be zero, and hence, these rules do not have to be analyzed. The fuzzy inference engine uses the appropriately designed knowledge base to evaluate the fuzzy rules and produce an output for each rule. Subsequently, the multiple outputs are transformed to a crisp output by the defuzzification interface.

Once the aggregated fuzzy set representing the fuzzy output variable for each subcontroller has been determined, an actual crisp control decision must be made. The process of decoding the output to produce an actual value for the control signal is referred to as defuzzification. Thus, a fuzzy logic controller-based center-average defuzzifier is implemented. The output of each fuzzy subcontroller is given by

$$f(e, \mathrm{Du}) = \frac{\sum_{l=1}^{9} \mu_{O^l} \left(\min_{i=1,2} \left(\mu_{F_i^l}(e, \mathrm{Du}) \right) \right)}{\sum_{l=1}^{9} \left(\min_{i=1,2} \left(\mu_{F_i^l}(e, \mathrm{Du}) \right) \right)} \tag{2}$$

which can be written in the form

$$f(e, \mathrm{Du}) = \theta^T \xi(e, \mathrm{Du}) = \xi^T(e, \mathrm{Du})\theta \tag{3}$$

where $\theta = (\mu_{O^1}, \ldots, \mu_{O^9})$, representing the degree of membership of the output

$$\xi^l(e, \mathrm{Du}) = \frac{\min_{i=1,2} \left(\mu_{F_i^l}(e, \mathrm{Du}) \right)}{\sum_{l=1}^{9} \left(\min_{i=1,2} \left(\mu_{F_i^l}(e, \mathrm{Du}) \right) \right)}. \tag{4}$$

The composed output of fuzzy PID was derived as follows:

$$u_{\mathrm{FPID}}(k) = u_{\mathrm{FP}}(k) + u_{\mathrm{FI}}(k) + u_{\mathrm{FD}}(k). \tag{5}$$

The control signal of the fuzzy P subcontroller

$$u_{\mathrm{FP}}(k) = f_P(e, \mathrm{Du}_P) G_P e(k). \tag{6}$$

The control signal of the fuzzy I subcontroller

$$u_{\mathrm{FI}}(k) = f_I(e, \mathrm{Du}_I) G_I \sum_{i=0}^{k} e(i)\, \Delta t, \qquad k = 0, 1, 2, \ldots. \tag{7}$$

The control signal of the fuzzy D subcontroller

$$u_{\mathrm{FD}}(k) = f_D(e, \mathrm{Du}_D)\, G_D\, \Delta e(k)/\Delta t \tag{8}$$

where G_P, G_I, and G_D are the scaling factors to the delayed control signals. The scaling factors are computed experimentally on a trial and error basis. The tuning parameters FK_P, FK_I and FK_D are provided by the output of each fuzzy inference mechanism. Therefore, the three apparent outputs of the fuzzy subcontrollers will be

$$\mathrm{FK}_P = f_p(e, \mathrm{Du}_P), \; \mathrm{FK}_I = f_I(e, \mathrm{DU}_I), \; \mathrm{FK}_D = f_D(e, \mathrm{DU}_D).$$

Saturation is also incorporated in the controller structure. As such, the final output of the fuzzy PID controller is

$$u(k) = \begin{array}{ll} u_{\max}(k) & u_{\mathrm{FPID}}(k) > u_{\max}(k) \\ u_{\mathrm{FPID}}(k) & u_{\min}(k) \leq u_{\mathrm{FPID}}(k) \leq u_{\max}(k) \\ u_{\min}(k) & u_{\mathrm{FPID}}(k) < u_{\min}(k) \end{array} \tag{9}$$

where $u_{\min}$ and $u_{\max}$ are the permitted minimum and maximum inputs to the brushless drive system.

III. Classical PID Versus Fuzzy PID Controller

The classical linear PID controller in its discrete form can be characterized as

$$u_{\mathrm{PID}} = K_P e(k) + K_I \sum_{i=0}^{k} e(i)\Delta t + K_D \Delta e(k)/\Delta t \qquad k = 0, 1, \ldots \tag{10}$$

where K_P, K_I, and K_D are constant gains.

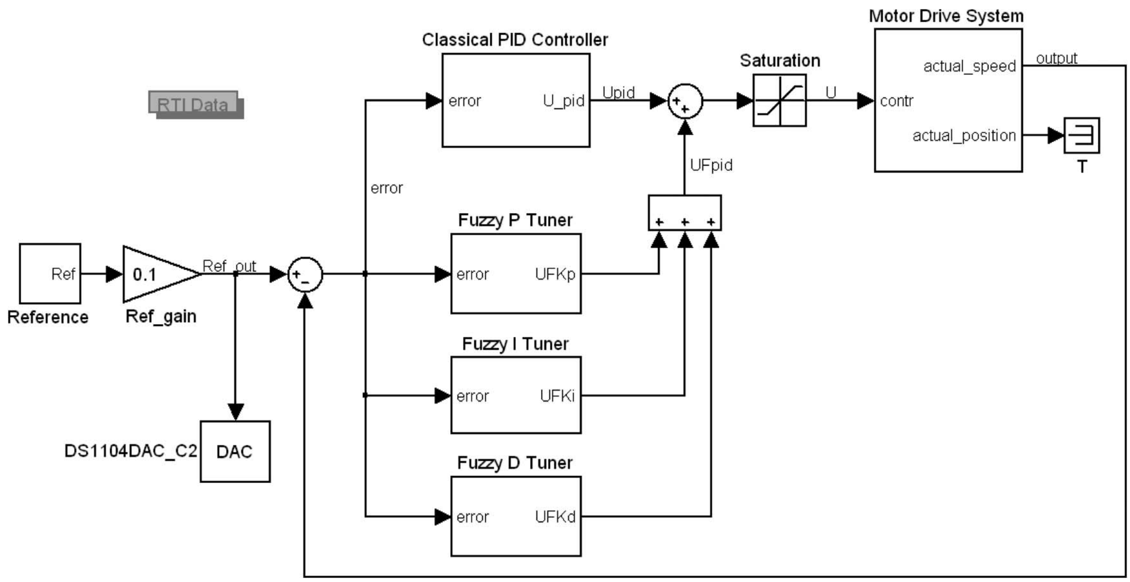


Fig. 5. Simulink model implementing the fuzzy PID controller scheme where all coefficients are tunable online.

Equation (5) provides a closed-form solution to the proposed fuzzy PID controller. It can be rewritten as

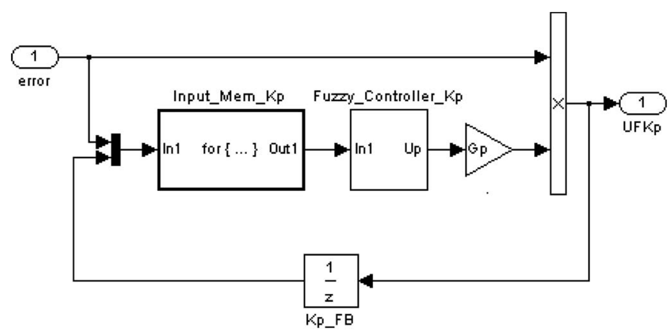
$$u_{\mathrm{FPID}}(k) = f_P(e, \mathrm{Du}_P)G_P\, e(k) + f_I(e, \mathrm{Du}_I)G_I \sum_{i=0}^{k} e(i)\Delta t + f_D(e, \mathrm{Du}_D)G_D\, \Delta e(k)/\Delta t \qquad k = 0, 1, 2, \ldots \tag{11a}$$

$$= (\mathrm{FK}_P)_{\mathrm{eq}}\, e(k) + (\mathrm{FK}_I)_{\mathrm{eq}} \sum_{i=0}^{k} e(i)\Delta t + (\mathrm{FK}_D)_{\mathrm{eq}}\, \Delta e(k)/\Delta t \qquad k = 0, 1, 2, \ldots \tag{11b}$$

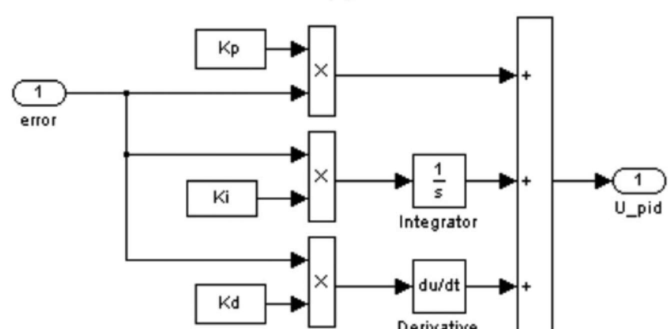
where $(\mathrm{FK}_P)_{\mathrm{eq}}$, $(\mathrm{FK}_I)_{\mathrm{eq}}$, and $(\mathrm{FK}_D)_{\mathrm{eq}}$ are defined to be the equivalent proportional, integral, and derivative gains to a classical linear PID controller, respectively. Comparing (11) with (10), we can conclude that parallel fuzzy PID controller (11) is a nonlinear PID controller with variable gains.

IV. Rapid Control Prototyping

The combination of dSPACE DS1104 DSP and MATLAB/Simulink/Real-Time Workshop (RTW) effectively created a rapid control prototype environment in which the designer focused on control design rather than programming details or debugging control languages. The design and implementation phases have been carried out using MATLAB/Simulink environment [16] and the commercial real-time hardware dSPACE DS1104 DSP-based DAC system. To design the controller, the designer must download a Simulink template file and connect the signals describing the reference, output, and command signals with suitable Simulink blocks. Simulink provides a graphical user interface for building system models as block diagrams using click-and-drag mouse operation. The Simulink template file used to implement the fuzzy PID controller is shown in Fig. 5. The parallel characteristics of the control structure are clearly detailed in the control diagram. Each fuzzy subcontroller (tuner) performs its control task independently of each other to contribute to the overall tracking goal. Fig. 6(a) shows the structure of the Fuzzy-P-Tuner. It presents its contribution to the overall control signal. For example, in the case of the Fuzzy-P-Tuner, the fuzzy system output is multiplied by the scaling factor G_P and then by the error to obtain the control signal u_{FP}. From the viewpoint of input–output relationship, its structure is analogous to that of the classical PID controller. Once the structure is fixed, the nonlinear property of the fuzzy PID controller is uniquely determined. Fig. 6(b) shows the Simulink block diagram of the classic PID controller.



(a)



(b)

Fig. 6. (a) Simulink block diagram of the Fuzzy-P-Tuner. (b) Simulink block diagram of the classic PID controller.

Once the controller is built in Simulink's block diagram, the designer can utilize the MATLAB RTW routine that can automatically produce C code from Simulink block diagram. The C code generated by RTW is used with the commercial real-time hardware dSPACE DS 1104 DSP-board for real-time control. Then, the interface between Simulink and the dSPACE DS 1104 DSP allows the control algorithms to be run on the hardware of the DS1104. Upon the completion of this process, the designer can validate the design using the laboratory platform at the click of a mouse button.

The dSPACE approach has the advantage of building on the designer's MATLAB/Simulink skills gained from basic training. A feature of this environment is a simple user interface, which requires a basic knowledge of MATLAB/Simulink for designing the controller. While an experiment is running, it is possible to change controller parameters and reference signals at the click of a mouse button. The memory occupation of the MATLAB generated C code is less than 1000 kB, and the sampling frequency is 1 kHz.

Generally, dSPACE DSP is a rapid prototyping tool that provides a means for the rapid development and testing of control algorithms by the real-time control of an actual target system through a flexible extensible multiprocessor environment. Under this unique environment, control engineers may well perform computer simulations, evaluate the simulated response of a system, develop and verify the performance of traditional and intelligent control laws in a simulated mode, and then easily install the developed controllers to the hardware all within the same routine interface. The control designer has the ability to simulate a control system against a model of the actual target system, test and refine the control system, and then deploy the control system to the dSPACE DSP without directly coding any software. The controller implemented on the dSPACE DSP executes, gathers inputs, computes the algorithms, and commands the output to the connected actuators. For the drive system under study, we actually did control the actual target system because we could set it up in the laboratory. For any other system, the user could use a model because he/she does not have an actual system to control. Therefore, the system control work would be simulation based.

V. Hardware System Description

The hardware apparatus is composed of four major elements: a dSPACE DS1104 DSP board, a controlled process (servomotor coupled with PM dc generator as a dynamic load), a driving circuit, and a personal computer (PC). Fig. 7 shows a photo of the experimental apparatus. The dSPACE DS1104 DSP board [17] forms the core of the closed loop system. Aside from the duties of controlling the operator interface, it performs the acquisition of the feedback signal, computes an error signal, delivers the error signal to the control algorithm, and executes the control algorithm to determine a control signal. The control algorithm is built within Simulink environment combined with the Real-Time Interface provided by dSPACE and is implemented by the main processor of the DS1104 board in real time. The motor is a 1-hp 3000-r/min three-phase brushless dc servomotor, which was manufactured by Moog Aerospace

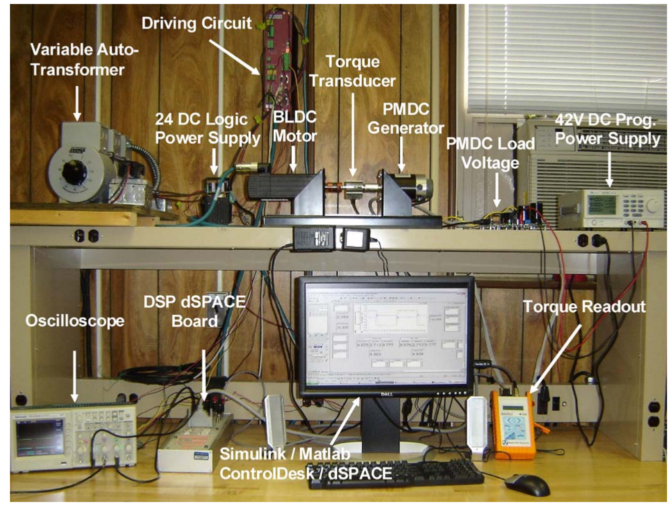


Fig. 7. Photograph of the experimental apparatus.

[18]. It is equipped with a resolver and coupled via a torque transducer. The motor is also coupled with a PM dc generator as a dynamic load. The driving board is also a Moog T200-410 designed for brushless servo drives [19].

The T200-410 driver board is interfaced with the PC through the DS1104 board. It provides two analog outputs points (TP1 and TP2) whereby process variables can be monitored. After programming the driving board, the actual position, and the actual speed can be monitored and serve as a source of feedback signals for the control algorithms. A low-pass third-order Butterworth filter is used for noise elimination, and the cutoff frequency is 5 rad/s. A variable autotransformer is used to supply the driving circuit with ac voltage of 230 V. A power supply is also used to supply the inverter component of the driving circuit with 24-V dc. The PC is a Pentium D 2.8-GHz PC with Windows XP.

VI. Experimental Implementation

In order to evaluate the performance of the proposed fuzzy PID controller, several test cases were completed in the laboratory under different operating conditions. The tests were repeated with the classical PID controllers under the same conditions. The standard PID controller is designed in accordance with the Ziegler–Nichols tuning criteria. The best gains were found experimentally to be $K_P = 0.2$, $K_I = 4$, and $K_D = 0.04$. Both fuzzy and classical PID controllers were designed such that the closed-loop control system would be stable and would meet given specifications associated with the following [4]:

1) stability robustness;
2) tracking performance at transient, including rise time, overshoot, and settling time;
3) regulation performance at steady state, including load disturbance rejection;
4) robustness against environmental uncertainty.

We operated the hardware and observed the closed loop performance of the proposed fuzzy PID controller. In order to compare tracking accuracy, the measured speed is placed over

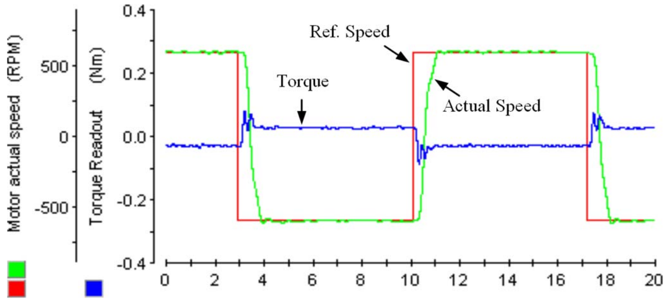


Fig. 8. Fuzzy PID speed tracking. $X = 1$ s/div, $Y1 = 0.1$ N · m/div, and $Y2 = 250$ r/min/div.

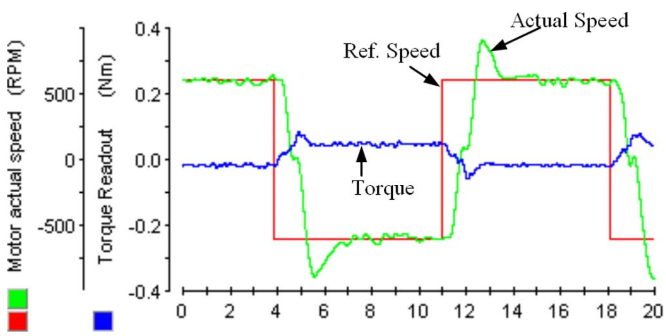


Fig. 9. PID speed tracking with antiwindup. $X = 1$ s/div, $Y1 = 0.1$ N · m/div, and $Y2 = 250$ r/min/div.

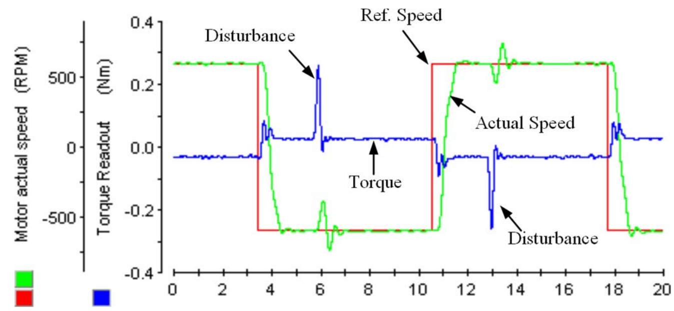


Fig. 10. Fuzzy PID speed tracking under disturbance. $X = 1$ s/div, $Y1 = 0.1$ N · m/div, and $Y2 = 250$ r/min/div.

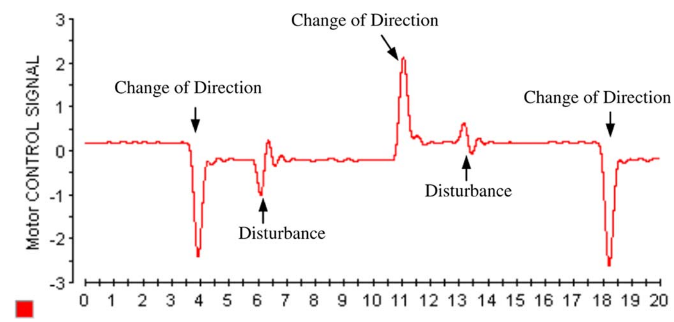


Fig. 11. PID speed tracking with antiwindup under disturbance. $X = 1$ s/div, $Y1 = 0.1$ N · m/div, and $Y2 = 250$ r/min/div.

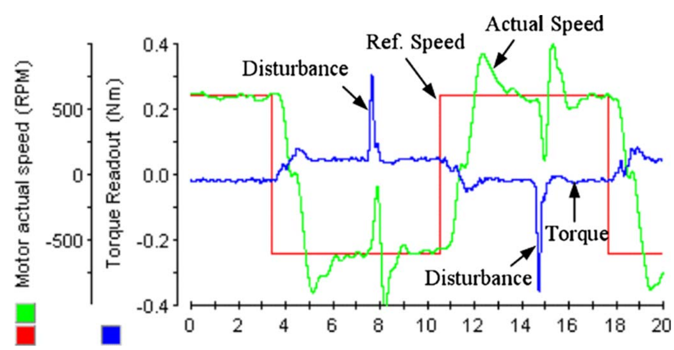


Fig. 12. Control effort for square-wave under disturbance. $X = 0.5$ s/div and $Y = 0.5$ V/div.

the specified desired reference speed. Figs. 8 and 9 show the speed tracking for a square-wave reference track with amplitude of 600 r/min. It is observed that the fuzzy PID controller brings the measured speed to the specified desired speed smoothly. It is also observed that the fuzzy PID controller eliminates the overshoot and steady-state error experienced when the classical PID controller is utilized.

The Simulink model implementing the PID controller has been customized to include an antiwindup mechanism. The source of the feedback to the integrator is the difference of the input to the saturation block from the output of the saturation block. When "windup occurs" and the control signal becomes larger than the saturation limit, the difference is negative. This negative value is passed through a gain block before arriving as feedback to the integrator. The negative value decreases the integral gain and thus reduces the main source of the "windup." As shown in Fig. 9, the classical PID controller with antiwindup scheme reduces the overshoot and extent of oscillations with trivial steady-state error under the same operating conditions. In both cases (Figs. 8 and 9), however, the torque is observed to spike when the speed is changing direction from one level to another at $t = 4, 11,$ and 18 s.

To demonstrate the capability and robustness of the proposed fuzzy PID controller, external disturbance was randomly applied to the servo drive system while the experiment was in progress. Figs. 10 and 11 show the performance of both controllers in the presence of external and sudden disturbance. Fig. 10 shows the fuzzy PID speed tracking under random load disturbance. Sudden load changes are applied at approximately 7 and 13 s. Fig. 10 also shows the variation in the torque results for speed tracking under the sudden load changes. The torque is observed to swing during each of the disturbances ($t = 7$ and $t = 13$ s). The torque is also observed to oscillate when the speed is changing direction from one level to another at $t = 4, 11,$ and 18 s. It can be observed that, the fuzzy PID controller rejects the disturbance effectively and continues to track the speed under these conditions. Fig. 11 shows the testing results for a classical PID controller under external load disturbances at instances shown on the figure. It is observed that the controller recovers from the disturbances but with considerable overshoot and sizeable steady-state error. It can also be seen that the undesirable oscillatory response is clearly evident. Fig. 12 shows an indication of how well the fuzzy PID controller succeeds in generating a voltage sequence (control signal) that forces the measured speed of the drive system to follow the specified reference trajectory. This is an indication of how well the control process is in proceedings during the experimental implementation.

To illustrate the effectiveness of the fuzzy PID controller further, the controller was applied to control the drive system

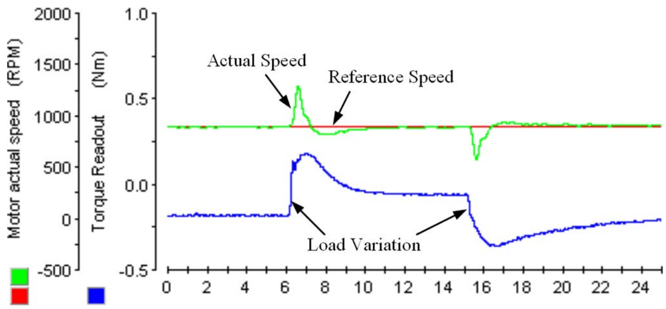


Fig. 13. Fuzzy PID speed tracking at constant speed. $X = 1$ s/div, $Y1 = 0.25$ N · m/div, and $Y2 = 250$ r/min/div.

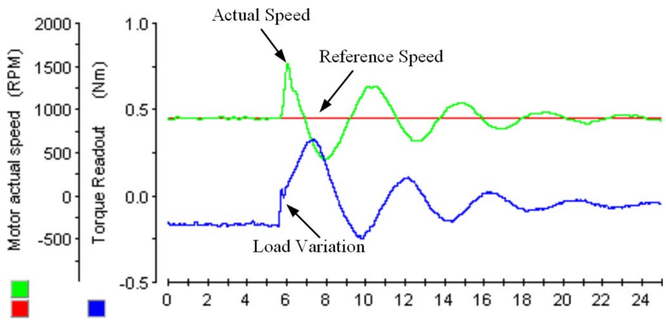


Fig. 14. PID speed tracking at constant speed. $X = 1$ s/div, $Y1 = 0.25$ N · m/div, and $Y2 = 250$ r/min/div.

under variable load torque. Fig. 13 shows the fuzzy PID speed regulation at 900 r/min. The load variations can be seen at approximately $t = 6$ and $t = 16$ s. The fuzzy PID recovers nicely from the load variations. It is observed that the fuzzy PID controller closely tracks the motor speed, even under changing conditions. Also, it is noticed that under variable load, the fuzzy PID controller brings smoothly the motor speed to the preset speed of 900 r/min. Sustaining load variations is also achieved. Fig. 13 also shows an indication of how well the torque transducer succeeds in generating the torque trajectory under load variations. The torque is observed to fluctuate at the occurrence of the load variations. The aforementioned test was repeated with a conventional PID controller running under the same conditions. The corresponding trajectory of the speed regulation with variable load torque is shown in Fig. 14. It can be seen that the classical PID controller causes an overshoot and some oscillation around the steady-state values in the output.

To display the capability of the fuzzy PID at zero speed performance, the motor speed operates at speed of 0 r/min and proceeds to desired speed of 600 r/min. The desired speed is then changed to 0 r/min. This was done to monitor the performance change of the drive system from standstill to medium speed. Fig. 15 shows the performance of the fuzzy PID controller under this operating condition. The top subplot displays the speed tracking at 0 and 600-r/min step, whereas the bottom subplot displays the variation in the torque performance for a square-wave speed tracking. The torque is observed to oscillate when the speed is changing from one level to another at approximately $t = 9$ and $t = 19$ s. Clearly, the rotor speed closely tracks the desired reference speed even at zero speed. Fig. 16 shows the speed tracking of the PID controller at zero

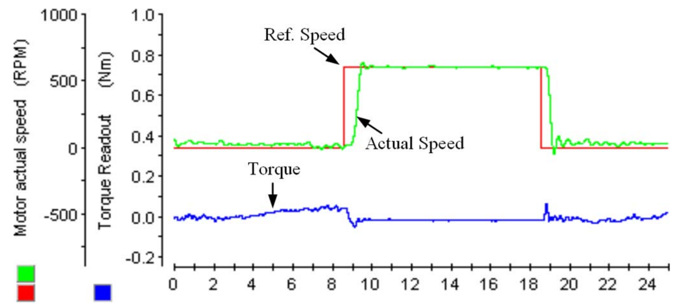


Fig. 15. Fuzzy PID speed tracking at zero speed. $X = 1$ s/div, $Y1 = 0.1$ N · m/div, and $Y2 = 250$ r/min/div.

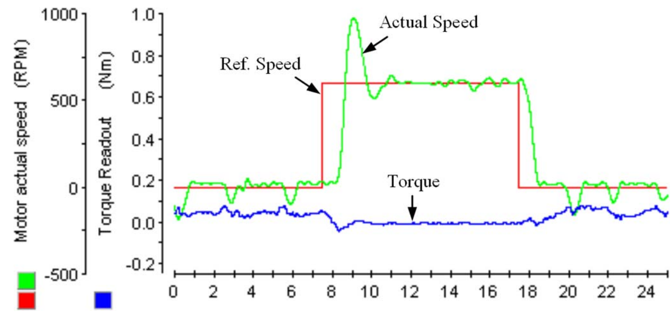


Fig. 16. PID speed tracking at zero speed. $X = 1$ s/div, $Y1 = 0.1$ N · m/div, and $Y2 = 250$ r/min/div.

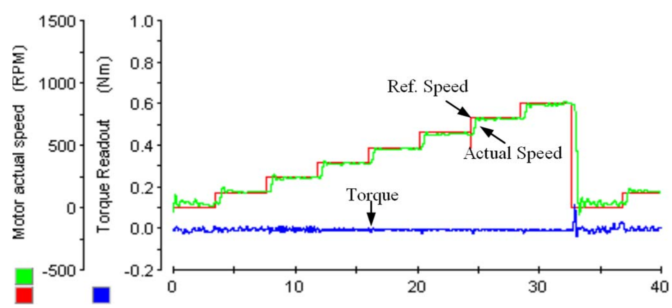


Fig. 17. Fuzzy PID speed tracking for step command signal. $X = 1$ s/div, $Y1 = 0.025$ N · m/div, and $Y2 = 250$ r/min/div.

speed. As expected, the conventional PID controller exhibits overshoot and some oscillation around the steady state.

Zero-to-high speed performance was also evaluated. In this test, the desired speed was changed from one arbitrary level to another. Fig. 17 shows an attractive speed reference signal that applied to the drive system. The top subplot displays the result of step command signal from zero- to high-speed. These step changes suggest a sudden load being applied to the drive system. From the top subplot, it can be observed that the drive system was able to track the speed under changing speed command. The bottom subplot displays the torque response for the step command trajectory. The torque response shows little oscillations at zero and low speed. The corresponding control effort for this tracking is shown in Fig. 18. This figure shows how well the controller succeeds in generating a voltage control sequence that forces the actual speed to follow the desired reference trajectory. Fig. 19 shows the speed tracking of the classical PID controller under the same conditions. As expected, the PID controller displays an overshoot and obvious oscillations just about zero and low speed. These results indicate that the fuzzy PID controller attains practical and stable control and

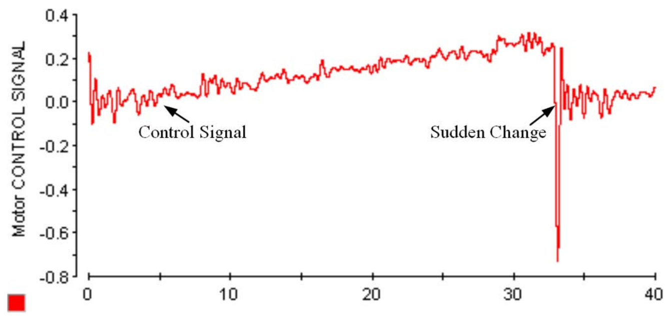


Fig. 18. Control effort for step-command signal under disturbance. $X = 5$ s/div and $Y1 = 0.1$ V/div.

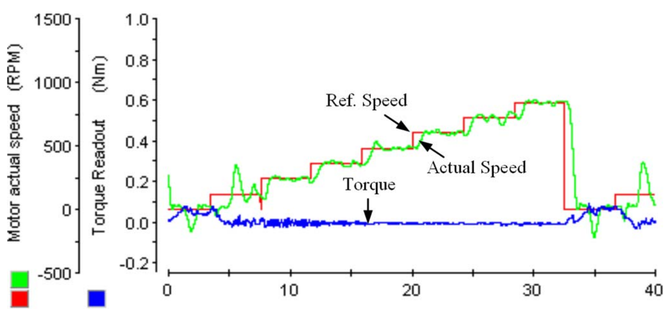


Fig. 19. PID speed tracking for step command signal. $X = 5$ s/div, $Y1 = 0.1$ N · m/div, and $Y2 = 250$ r/min/div.

is sufficiently robust to yield accurate tracking performance, regardless of the types of desired reference trajectories.

VII. Conclusion

A robust fuzzy PID controller and its integration with a classical PID in a global control scheme has been developed and described in this paper. The resulting fuzzy PID controller has been implemented and demonstrated in the laboratory, and its effectiveness in tracking application has also been verified. Experimental results have shown excellent tracking performance of the proposed fuzzy PID controller and have convincingly demonstrated the usefulness of the proposed fuzzy PID controller in motor drives with uncertainties. The efficacy of the fuzzy PID controller has been demonstrated by its positive results, when compared with those of the classical PID controller. The laboratory implementation was based on dSPACE DS1104 DSP and MATLAB/Simulink environment. This environment allowed for extensive experimentation, performance comparison, and development of an improved global control scheme. The combination of dSPACE DS1104 DSP and MATLAB/Simulink/RTW effectively created a rapid control prototype environment, in which the designer focused on control design rather than programming details or debugging control languages. In this way, the environment offers an unparalleled experience and is a great source of attracting motion control engineers and practitioners and exciting their interest.

References

[1] G. J. Silva, A. Datta, and S. P. Bhattacharyya, "New results on the synthesis of PID controllers," *IEEE Trans. Autom. Control*, vol. 47, no. 2, pp. 241–252, Feb. 2002.

[2] S. Parvez and Z. Gao, "A wavelet-based multiresolution PID controller," *IEEE Trans. Ind. Appl.*, vol. 41, no. 2, pp. 537–543, Mar./Apr. 2005.

[3] M. Tursini, F. Parasiliti, and D. Zhang, "Real-time gain tuning of PI controllers for high-performance PMSM drives," *IEEE Trans. Ind. Appl.*, vol. 38, no. 4, pp. 1018–1026, Jul./Aug. 2002.

[4] K. H. Ang, G. Chong, and Y. Li, "PID control system analysis, design, and technology," *IEEE Trans. Control Syst. Technol.*, vol. 13, no. 4, pp. 559–576, Jul. 2005.

[5] K. Kim and R. C. Schaefer, "Tuning a PID controller for a digital excitation control system," *IEEE Trans. Ind. Appl.*, vol. 41, no. 2, pp. 485–492, Mar./Apr. 2005.

[6] C. Knope, "PID control,"*IEEE Control Syst. Mag.,* vol. 26, no. 1, pp. 30–31, Feb. 2006.

[7] Y. Li, K. H. Ang, and G. Chong, "Patents, software, and hardware for PID control," *IEEE Control Syst. Mag.*, vol. 26, no. 1, pp. 42–54, Feb. 2006.

[8] H.-X. Li, L. Zhang, K.-Y. Cai, and G. Chen, "An improved robust fuzzy-PID controller with optimal fuzzy reasoning," *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 35, no. 6, pp. 1283–1294, Dec. 2006.

[9] S. Skoczowski, S. Domek, K. Pietrusewicz, and B. Plater, "A method for improving the robustness of PID control," *IEEE Trans. Ind. Electron.*, vol. 52, no. 6, pp. 1669–1676, Dec. 2005.

[10] B.-G. Hu, G. K. Mann, and R. G. Gosine, "A systematic study of fuzzy PID controllers—function-based evaluation approach," *IEEE Trans. Fuzzy Syst.*, vol. 9, no. 5, pp. 699–712, Oct. 2001.

[11] C. W. Tao and J.-S. Taur, "Flexible complexity reduced PID-like fuzzy controllers," *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 30, no. 4, pp. 510–516, Aug. 2000.

[12] J.-X. 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.

[13] B.-G. Hu, G. K. Mann, and R. G. Gosine, "New methodology for analytical and optimal design of fuzzy PID controllers," *IEEE Trans. Fuzzy Syst.*, vol. 7, no. 5, pp. 521–538, Oct. 1999.

[14] M. Cheng, Q. Sun, and E. Zhou, "New self-tuning fuzzy PI control of a novel doubly salient permanent-magnet motor drive," *IEEE Trans. Ind. Electron.*, vol. 53, no. 3, pp. 814–821, Jun. 2006.

[15] A. Rubaai, D. Ricketts, and M. D. Kankam, "Laboratory implementation of a microprocessor-based fuzzy logic tracking controller for motion controls and drives," *IEEE Trans. Ind., Appl.*, vol. 38, no. 2, pp. 448–456, Mar./Apr. 2002.

[16] *MATLAB/Simulink User's Guide*, The MathWorks Inc., Natick, MA, 1998.

[17] *dSPACE User's Guide*, dSPACE Digital Signal Process. Control Eng. GMBH, Paderborn, Germany, 2003.

[18] *G413-817 Technical Data Manual*, Moog Aerospace, East Aurora, NY, 2000.

[19] *T200-410 Technical Data Manual*, Moog Aerospace, East Aurora, NY, 2000.

Ahmed Rubaai (S'88–M'88–SM'03) received the M.S.E.E. degree from Case Western Reserve University, Cleveland, OH, in 1983, and the Dr.Eng. degree from Cleveland State University, Cleveland, in 1988.

Since 1988, he has been with Howard University, Washington, DC, as a Faculty Member, where he is currently a Professor of electrical engineering in the Electrical and Computer Engineering Department. He is the Founder and Lead Developer of the new Multipurpose and Multidisciplinary Control System Laboratory, Howard University, and is actively involved in many projects with industry, while engaged in teaching, research, and consulting in the area of artificial intelligence. His research interests include high-performance motor drives, research and development of intelligent applications for manufacturing systems, and industrial applications.

Dr. Rubaai was the recipient of the IEEE Industry Applications Society (IAS) Second Prize Paper Award, in September 2007, American Society for Engineering Education (ASEE) Division of Experimentation and Laboratory Oriented Studies Best Paper Award, in June 2006, and the IEEE/IAS Honorable Mention Prize Paper Award, in October 2002. He was also the recipient of the Howard University Exemplary Teaching Award, in April 2005. Prior to winning this coveted award, he was named "Professor of the Year," in 1997 and 1998 by Howard's School of Engineering. In April 2001, he received the ASEE Middle Atlantic Section Distinguished Educator Award. In June 2004, he received a Software Release Award from NASA Glenn Invention and Contribution Division. He is a past Chair of the Industrial Automation and Control Committee of the IEEE/IAS. Currently, he is an Executive Board Member of the IEEE/IAS and Chair of the IEEE/IAS Manufacturing Systems Development and Applications Department.

Marcel J. Castro-Sitiriche (S'05) was born in Santurce, Puerto Rico, in 1978. He received the B.Sc. degree in electrical engineering from the University of Puerto Rico, Mayagüez, PR, in 2000, and the Ph.D. degree in electrical engineering from Howard University, Washington, DC, in 2007.

Since 2007, he has been with Howard University as a Faculty Member in the Electrical and Computer Engineering Department. His current research interests are in the areas of artificial intelligence, electric motor drives, and control systems.

Abdul R. Ofoli (S'02–M'07) received the B.Sc. degree in electrical and electronic engineering from Kwame Nkrumah University of Science and Technology, Kumasi, Ghana, in 1999, and the M.Eng. and Ph.D. degrees in electrical engineering from Howard University, Washington, DC, in 2002 and 2006, respectively.

Since 2006, he has been with Cummins Inc., Columbus, IN, where he is currently a Control Engineer. His research interests are in the areas of intelligent control of power systems and power electronics devices with fuzzy logic controllers and artificial neural networks, fault diagnosis, microprocessor applications, and industrial networks.