Research Article

ModelSim/Simulink Cosimulation and FPGA Realization of a Multiaxis Motion Controller

Ying-Shieh Kung,¹ Jin-Mu Lin,¹ Yu-Jen Chen,¹ and Hsin-Hung Chou²

¹Department of Electrical Engineering, Southern Taiwan University of Science and Technology, 1 Nan-Tai Street, Yong-Kang District, Tainan City 710, Taiwan

²Mechanical and System Research Laboratories, Industrial Technology Research Institute, Section 4, 195 Chung Hsing Road, Chutung, Hsinchu 310, Taiwan

Correspondence should be addressed to Ying-Shieh Kung; kung@mail.stust.edu.tw

Received 20 August 2014; Accepted 13 September 2014

Academic Editor: Stephen D. Prior

Copyright © 2015 Ying-Shieh Kung et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

This paper is to implement a multiaxis servo controller and a motion trajectory planning within one chip. At first, SoPC (system on a programmable chip) technology which is composed of an Altera FPGA (field programmable gate arrays) chip and an embedded soft-core Nios II processor is taken as the development of a multiaxis motion control IC. The multiaxis motion control IC has two modules. The first module is Nios II processor which realizes the motion trajectory planning by software. It includes the step, circular, window, star, and helical motion trajectory. The second module presents a function of the multiaxis position/speed/current controller IP (intellectual property) by hardware. And VHDL (VHSIC Hardware Description Language) is applied to describe the multiaxis servo controller behavior. Before the FPGA realization, a cosimulation work by ModelSim/Simulink is applied to test the VHDL code. Then, this IP combined by Nios II processor will be downloaded to FPGA. Therefore, a fully digital multiaxis motion controller can be realized by a single FPGA chip. Finally, to verify the effectiveness and correctness of the proposed multiaxis motion control IP, a three-axis motion platform (XYZ table) is constructed and some experimental results are presented.

1. Introduction

The integration of a multiaxis servo and motion controller into one chip and then application to XYZ table, CNC (Computed Numerically Controlled) machine or robotic system, and so forth, for reducing the system volume and increasing the system performance and motion stability, have become a popular research topic in literatures [1–5]. Further, in typical applications, adopting PMSM as an actuator in automatic multiaxis equipment is a major technique trend due to its excellent high power density and better motion control characteristics, fast response and accuracy positioning [6–8]. In traditional machine, it usually utilizes one central controller for planning the motion trajectory and multiple microprocessors for multiaxis servo motor drive system to establish a multiaxis motion controller. In this architecture, the coordination and communication between center controller and multiple microprocessors are a challenge task in the automatic multiaxis equipment. However,

some drawbacks appear, such as large volume, being easily affected by the noise, expensive cost, and inflexible. In addition, data communication and handshaking protocol between the central controller and multiple microprocessors certainly slow down the system executing speed. To solve this problem, FPGA provides a best solution among those commercial high-end microprocessors due to its fast computational power, parallel processing ability, field programmable characteristics, and SoPC environment which can perform hardware-software codesign task [9–12]. For example, in the design of PMSM servo drive using FPGA, it only needs less than $2\ \mu\text{s}$ in realizing the SVPWM algorithm, less than $3\ \mu\text{s}$ in implementing the current vector controller, and less than $3\ \mu\text{s}$ in executing the fuzzy controller algorithm in speed loop. The fast computational and parallel processing characteristics make it possible to integrate multiaxis servo controllers and the function of motion trajectory planning into one FPGA.

Recently, a cosimulation work by Electronic Design Automation (EDA) Simulator Link has been gradually

applied to verify the effectiveness of the HDL (Hardware Description Language) code in the motor drive system [13–15]. The EDA Simulator Link [16] provides a cosimulation interface between Simulink and ModelSim [17] where you can use Simulink models as a test bench that generates stimulus for ModelSim and analyzes the simulation's response [16]. Therefore, before the FPGA realization, a cosimulation work by EDA Simulator Link will be applied to the multiaxis motion control system in this paper. The multi-PMSMs, multi-inverters, a motion trajectory planning, and motion position response are performed in Simulink and the position/speed/current controller IP (Intellectual Property) which is described by VHDL code is executed in ModelSim. After successful verification in simulation, the position/speed/current controller IP can be directly used in the FPGA.

The realization in this paper is the SoPC technology in which FPGA embedding a Nios II processor is established to develop a multiaxis motion control IC, and it is shown in Figure 1. It includes multiple servo controllers IP and one Nios II processor in a FPGA. Each servo controller IP performs the function of position/speed/current controller for each PMSM drive and Nios II processor executes the function of motion trajectory planning. In addition, the detailed inside architecture of the proposed multiaxis motion control IC in FPGA is presented in Figure 2. In servo controller IP, the vector control is adopted in the current loop, P controller is used in the speed loop, and fuzzy controller is applied in the position loop for each PMSM drive. However, the multiaxis motion control IC shown in Figure 1 can be applied multiaxis automatic equipment, such as XYZ table, SCARA robot arm, vertical-type robot arm, and CNC machine.

2. Modelling, Control, and Motion Trajectory Design

Figure 2 includes multiple servo controllers IPs and one Nios II processor in a FPGA. Each servo controller IP performs the function of position/speed/current controller for one PMSM drive and Nios II processor executes the function of motion trajectory planning. Detailed design methodology is described as follows.

2.1. Mathematical Model of Single-Axis Table. The typical mathematical model of a PMSM is described, in two-axis d - q synchronous rotating reference frame, as follows:

$$\begin{aligned} \frac{di_d}{dt} &= -\frac{r_s}{L_d} i_d + \omega_e \frac{L_q}{L_d} i_q + \frac{1}{L_d} v_d, \\ \frac{di_q}{dt} &= -\omega_e \frac{L_d}{L_q} i_d - \frac{r_s}{L_q} i_q - \omega_e \frac{\lambda_f}{L_q} + \frac{1}{L_q} v_q, \end{aligned} \quad (1)$$

where v_d, v_q are the d - and q -axis voltages; i_d, i_q are the d - and q -axis currents; r_s is the phase winding resistance; L_d, L_q are the d - and q -axis inductance; ω_e is the rotating speed of magnet flux; λ_f is the permanent magnet flux linkage.

The current loop control of PMSM drive in Figure 2 is based on a vector control approach. That is, if the i_d is

controlled to 0, the PMSM will be decoupled and controlling a PMSM will be similar to controlling a DC motor. Therefore, after decoupling, the torque of PMSM can be written as the following equation:

$$T_e = \frac{3P}{4} \lambda_f i_q \triangleq K_t i_q \quad (2)$$

with

$$K_t = \frac{3P}{4} \lambda_f. \quad (3)$$

T_e is the motor torque and P is pole pairs. Thus, considering the mechanical load, the overall dynamic equation of linear table system is obtained by

$$T_e - T_L = J_m \frac{2\pi}{r_b} \frac{d^2 s_{pi}}{dt^2} + B_m \frac{2\pi}{r_b} \frac{ds_{pi}}{dt}. \quad (4)$$

K_t is force constant; J_m is the inertial value; B_m is damping ratio; T_L is the external torque; s_{pi} represents the displacement in i -axis table; r_b is the lead of the ball screw.

2.2. Fuzzy Controller (FC) Design. The FC in this study uses singleton fuzzifier, triangular membership function, product-inference rule, and central average defuzzifier method. In Figure 2, the tracking error e and the error change de are defined by

$$\begin{aligned} e(k) &= s_{pi}^*(k) - s_{pi}(k), \\ de(k) &= e(k) - e(k-1), \end{aligned} \quad (5)$$

where u_f represents the output of the FC; $S_{pi}^*(k)$, $S_{pi}(k)$ represent i th-axis position command and i th-axis position feedback, respectively. The design procedure of the FC is described as follows.

(a) Take e, de as the input variable of FC, and define their linguist variables as E and dE . The linguist values of E and dE are $\{A_0, A_1, A_2, A_3, A_4, A_5, A_6\}$ and $\{B_0, B_1, B_2, B_3, B_4, B_5, B_6\}$, respectively. Each linguist value of E and dE is based on the symmetrical triangular membership function.

(b) Compute the membership degree of e and de . It is only two linguistic values which are excited (resulting in a nonzero membership) in any input value. Therefore, the membership degree $\mu_{A_i}(e)$ can be straightforwardly obtained by

$$\mu_{A_i}(e) = \frac{e_{i+1} - e}{2}, \quad \mu_{A_{i+1}}(e) = 1 - \mu_{A_i}(e) \quad (6)$$

and $e_{i+1} \triangleq -6 + 2^*(i + 1)$. Similar results can be obtained in computing the membership degree $\mu_{B_j}(de)$.

(c) Select the initial FC rules by referring to the dynamic response characteristics, such as

$$\begin{aligned} \text{IF } e \text{ is } A_i \text{ and } \Delta e \text{ is } B_j \text{ THEN } u_f \text{ is } c_{ji}, \\ \text{for } i, j = 0 \sim 6, \end{aligned} \quad (7)$$

FPGA-based multiaxis motion control IC

FIGURE 1: FPGA-based multiaxis motion control IC and its applications.

where A_i and B_j are fuzzy number and $c_{j,i}$ is real number.

- (d) Establish the fuzzy system $u_f(e, de)$ by using the singleton fuzzifier, product-inference rule, and central average defuzzifier method. Although there are total 49 fuzzy rules that will be inferred, actually only 4 fuzzy rules can be effectively excited to generate a nonzero output. Therefore the output of the fuzzy inference can be obtained by the following expression:

$$u_f(e, de) = \frac{\sum_{n=i}^{i+1} \sum_{m=j}^{j+1} c_{m,n} [\mu_{A_n}(e) * \mu_{B_m}(de)]}{\sum_{n=i}^{i+1} \sum_{m=j}^{j+1} \mu_{A_n}(e) * \mu_{B_m}(de)} \quad (8)$$

$$\triangleq \sum_{n=i}^{i+1} \sum_{m=j}^{j+1} c_{m,n} * d_{n,m},$$

where $d_{n,m} \triangleq \mu_{A_n}(e) * \mu_{B_m}(de)$ and $\sum_{n=i}^{i+1} \sum_{m=j}^{j+1} d_{n,m} = 1$.

2.3. Motion Trajectory Planning. The circular, window, star, and helical circular motion trajectories are typically used [18] as the performance evaluation of the motion controller for XYZ table.

- (a) Circular motion trajectory: it is computed as follows:

$$\begin{aligned} x_i &= r \sin(\theta_i), \\ y_i &= r \cos(\theta_i) \end{aligned} \quad (9)$$

with $\theta_i = \theta_{i-1} + \Delta\theta$, where $\Delta\theta$, r , x_i , y_i are angle increment, radius, x-axis trajectory command, and y-axis trajectory command, respectively.

- (b) Window motion trajectory: it is shown in Figure 3(a). The formulation is derived as follows:

a-trajectory:

$$x_i = x_{i-1}, \quad y_i = S + y_{i-1}, \quad (10)$$

b-trajectory:

$$\begin{aligned} & \left(\theta_i : \frac{6}{4}\pi \rightarrow 2\pi, \theta_i = \theta_{i-1} + \Delta\theta \right), \\ x_i &= O_{x1} + r \cos(\theta_i), \quad y_i = O_{y1} + r \sin(\theta_i), \end{aligned} \quad (11)$$

c-trajectory:

$$x_i = S + x_{i-1}, \quad y_i = y_{i-1}, \quad (12)$$

d-trajectory:

$$\begin{aligned} & \left(\theta_i : \pi \rightarrow \frac{6}{4}\pi, \theta_i = \theta_{i-1} + \Delta\theta \right) \\ x_i &= O_{x2} + r \cos(\theta_i), \quad y_i = O_{y2} + r \sin(\theta_i), \end{aligned} \quad (13)$$

e-trajectory:

$$x_i = x_{i-1}, \quad y_i = -S + y_{i-1}, \quad (14)$$

f-trajectory:

$$\begin{aligned} & \left(\theta_i : \frac{1}{2}\pi \rightarrow \pi, \theta_i = \theta_{i-1} + \Delta\theta \right) \\ x_i &= O_{x3} + r \cos(\theta_i), \quad y_i = O_{y3} + r \sin(\theta_i), \end{aligned} \quad (15)$$

FIGURE 2: The inside architecture of FPGA-based multi-axis position/speed/current control IC.

FIGURE 3: (a) Window motion trajectory. (b) Star motion trajectory.

g-trajectory:

$$x_i = -S + x_{i-1}, \quad y_i = y_{i-1}, \quad (16)$$

h-trajectory:

$$\left(\theta_i : 0 \rightarrow \frac{1}{2}\pi, \theta_i = \theta_{i-1} + \Delta\theta \right) \\ x_i = O_{x4} + r \cos(\theta_i), \quad y_i = O_{y4} + r \sin(\theta_i), \quad (17)$$

i-trajectory:

$$x_i = x_{i-1}, \quad y_i = S + y_{i-1}, \quad (18)$$

where S are position increment. In addition, the (O_{x1}, O_{y1}) , (O_{x2}, O_{y2}) , (O_{x3}, O_{y3}) , (O_{x4}, O_{y4}) are arc center of b-, d-, f-, and h-trajectory in Figure 3(a) and r is the radius. The motion speed of the table is determined by $\Delta\theta$.

(c) Star motion trajectory: it is shown in Figure 3(b). The formulation is derived as follows:

a-trajectory:

$$x_i = S + x_{i-1}, \quad y_i = y_{i-1}, \quad (19)$$

b-trajectory:

$$x_i = -S * \sin 54^\circ + x_{i-1}, \quad y_i = -S * \sin 36^\circ + y_{i-1}, \quad (20)$$

c-trajectory:

$$x_i = S * \sin 18^\circ + x_{i-1}, \quad y_i = S * \sin 72^\circ + y_{i-1}, \quad (21)$$

d-trajectory:

$$x_i = S * \sin 18^\circ + x_{i-1}, \quad y_i = -S * \sin 72^\circ + y_{i-1}, \quad (22)$$

e-trajectory:

$$x_i = -S * \sin 54^\circ + x_{i-1}, \quad y_i = S * \sin 36^\circ + y_{i-1}. \quad (23)$$

The motion speed of the table is determined by S .

(d) Helical circular motion trajectory: it is computed as follows:

$$x_i = r \sin(\theta_i) \\ y_i = r \cos(\theta_i) \\ z_i = z_{i-1} + \Delta z \quad (24)$$

with $\theta_i = \theta_{i-1} + \Delta\theta$, where $\Delta\theta$, Δz , r , x_i , y_i , z_i are angle increment, height increment, radius, x -axis trajectory command, y -axis trajectory command, and z -axis trajectory command, respectively.

3. Digital Circuit Design for Position/Speed/Current Controller IP

The proposed FPGA-based three-axis motion control IC for XYZ table is shown in Figure 4. In position/speed/current controller IP, the vector control is adopted in the current loop, P controller is used in the speed loop, and fuzzy controller is applied in the position loop. The FPGA herein uses Cyclone IV-EP4CE115, which is the product of Altera cooperation. There are 114,480 LEs (Logic Elements), 3,981,312 RAM bits, and 150 embedded multipliers in the Cyclone IV. In addition, a Nios II processor is embedded into FPGA to establish an SoC environment. The motion control IC comprises a Nios II embedded processor and multiple position/speed/current controller IP for multiaxis machine. The Nios II processor is utilized to perform the motion trajectory computation. The code in Nios processor is developed with C language. Main program establishes the initial parameters setting, the motion command setting, and so forth and the ISR (interrupt service routine) executes the motion trajectory computation, position command generation, and sending out to three controller IPs. Each internal digital circuit of position/speed/current controller IP is also shown in Figure 4 and it includes one circuit of position FC and speed P controller, one circuit of current controller and coordinate transformation (CCCT), one SVPWM, one ADC interface, and one QEP interface circuit. All circuits in Figure 4 are realized by hardware in FPGA, and the circuit behavior is described by VHDL. The sampling frequency in position, speed, and current control loop are designed with 2 kHz, 2 kHz, and 16 kHz, respectively. The operating clock of the designed FPGA controller is 50 MHz and the frequency divider generates another 12.5 MHz clock to supply all module circuits in Figure 4. A finite state machine (FSM) method is employed to model the algorithms of the position FC and speed P controller, the CCCT, and the SVPWM, which utilizes one adder, one multiplier, a look-up table, comparators, and registers to carry out the overall computation. It needs to spend 20 steps ($1.6 \mu s$), 24 steps ($1.92 \mu s$), and 18 steps ($1.44 \mu s$) to complete the computation of position/speed controller, CCCT, and SVPWM, respectively. The data type is designed with 16-bit length, 2's complement, and Q15 format in the position/speed controller and CCCT, but 12-bit length, 2's complement, and Q11 format in SVPWM. For detailed figures and description regarding CCCT and SVPWM refer to [4]. For an example of a three-axis motion control IC, overall circuit included a Nios II processor and three position/speed/current controllers IP consume 22,688 LEs (Logic Elements), 73 embedding multipliers, 241 IO pins, and 241,792 memory bits.

4. ModelSim/Simulink Cosimulation and Simulation Results

The Simulink/ModelSim cosimulation architecture for the three-axis motion control system is shown in Figure 5. Each axis system is driven by one PMSM. The SimPowerSystem blockset in the Simulink executes the three PMSMs

FIGURE 4: Internal circuit design of the proposed FPGA-based multi-axis motion control IC.

and three IGBT-based inverters. The EDA Simulator Link for ModelSim executes the cosimulation using VHDL code running in ModelSim program. It carries out the function of three position/speed/current controller IPs with six works in Figure 5. The work-1, work-3, and work-5 perform the function of position and speed loop controller, and work-2, work-4, and work-6 accomplish the function of current vector controller which includes current PI controller, coordinate transformation (CCCT), and space vector pulse width modulation (SVPWM). The signals of three-phase current, rotor speed, and rotor position in PMSM-1 are, respectively, fed back to work-3 and work-1 to do the close-loop position control. The similar conditions of close-loop position control in PMSM-2 and PMSM-3 drive system are the same. The sampling frequency in current control loop is designed with 16 kHz but in position/speed control loop is 1 kHz. The motion trajectory described in Section 2.3 is planned within the block diagram of the Matlab embedded function in

Figure 5. Matlab m-code is used to develop the program of the motion trajectory.

In this simulation cases, the following motion trajectory will be carried out and tested: (1) single-axis position step response; (2) circular motion trajectory tracking (two-axis simultaneous motion); (3) window motion trajectory tracking (two-axis simultaneous motion); (4) star motion trajectory tracking (two-axis simultaneous motion); (5) helical circular motion trajectory tracking (three-axis simultaneous motion). Firstly, single-axis position step response is considered, and the step signal with 1.5 s and 5 mm amplitude is tested. The simulation results in position response and current response are presented in Figure 6. In this simulation, the K_p and K_i gains in position controller are, respectively, chosen by (15000, 0) and (25000, 95) in 16bit-Q15 format. In Figure 6(a), the position responses with 0.03 seconds rising time and zero overshoot and zero which show a good response performance. In Figure 6(b),

FIGURE 5: Simulink/ModelSim cosimulation architecture for three-axis motion control system.

FIGURE 6: Single-axis PMSM: (a) step position response; (b) current response.

FIGURE 7: (a) Circular motion trajectory tracking; (b) tracking error; (c) position response; (d) current response.

FIGURE 8: (a) Window motion trajectory tracking; (b) tracking error; (c) position response; (d) current response.

the current in i_d axis approaches zero which presents the success of the vector control in PMSM drive. Further, in the two-axis simultaneous motion trajectory, the circular motion trajectory control with center (5, 5) mm and radius 10 mm is evaluated and the simulation results in motion trajectory tracking, tracking error, position response, and current response are shown in Figure 7. In this design, the K_p and K_i gains in position controller are, respectively, chosen by (18000, 0), (19000, 95) in 16bit-Q15 format. The tracking errors in Figure 7(b) show the maximum ± 0.18 mm

in x-axis table, maximum ± 0.18 mm in y-axis table, and less than 0.1855 mm in overall circular motion tracking. It demonstrates a good tracking. Additionally, the current response of i_d -axis both in x- and y-axis motors all can be controlled to zero which presents the success of the vector controller. In the window tracking motion, the trajectory is designed as Figure 3(a) and its simulation results in motion trajectory tracking, tracking error, position response, and current response are shown in Figure 8. In this design, the K_p and K_i gains in position controller are, respectively,

FIGURE 9: (a) Star motion trajectory tracking; (b) tracking error; (c) position response; (d) current response.

chosen by (15000, 0), (25000, 95) in 16bit-Q15 format. The tracking errors in Figure 8 show the maximum ± 0.185 mm in x -axis table, maximum ± 0.178 mm in y -axis table, and less than 0.19 mm in overall motion tracking. In the star tracking motion, the trajectory is designed as Figure 3(b) and its simulation results are shown in Figure 9. In this experiment, the K_p and K_i gains in position controller are, respectively, chosen by (15000, 0), (25000, 95) in 16bit-Q15 format. The tracking errors in Figure 9(b) show the maximum ± 0.14 mm in x -axis table, maximum ± 0.14 mm in y -axis table, and less than 0.17 mm in overall star motion

tracking. The results show a good tracking response in both window and star motion trajectory. Finally, the three-axis simultaneous motion trajectory which is helical circular trajectory is evaluated, and their simulation tracking results are shown in Figure 10, respectively. In this design, the K_p and K_i gains in position controller are, respectively, chosen by (18000, 0), (19000, 95) in 16bit-Q15 format. In Figure 10(b), the tracking errors show that it can keep within ± 0.185 mm in x - and y -axis table motion and ± 0.19 mm in z -axis table motion which reveals a good tracking response. Therefore, Figure 16 shows a good tracking result

FIGURE 10: (a) Helical circular motion trajectory and (b) tracking results.

FIGURE 11: Experimental system.

FIGURE 12: x -axis table: (a) position step response; (b) current response.

FIGURE 13: (a) Circular motion trajectory tracking; (b) tracking error; (c) position response; (d) current response.

in 3-axis motion tracking. Further, Figures 6~10 present that the three-axis motion control system tested through ModelSim/Simulink cosimulation is effectiveness and correctness. After confirming the success of the proposed three-axis position/speed/current controller IP, it will directly apply to FPGA to be further evaluated.

5. Experimental System and Results

5.1. Experimental System. The overall experimental system shown in Figure 11 includes an FPGA control board, three

sets of voltage source IPM-based (Intelligent Power Modules) inverter, and an XYZ table which is driven by three PMSMs and three ball-screws. The power, rating, voltage, current, and rating speed of PMSM are 200 W, 92 V, 1.6 A, and 3000 rpm, respectively. A 2500 ppr rotary encoder attached to PMSM is used to measure the motor's electrical angle. Each ball-screw has 5 mm lead. Inside IPM, it has 6 sets of IGBT type power transistors with emitter of the rated voltage of 600 V, collector DC current rating of 20 A, and a short time (1 ms) rated current of 40 A. Photo coupler IC adopts TLP250 whose output has push-pull amplifier functions. The FPGA uses

FIGURE 14: (a) Window motion trajectory tracking; (b) tracking error; (c) position response; (d) current response.

Cyclone IV-EP4CE115 which is the kernel of this system. There are 114,480 LEs (Logic Elements), 3,981,312 RAM bits, and 150 embedded multipliers in the Cyclone IV. Further, a Nios II processor can be embedded into FPGA to establish SoC environment.

5.2. Experimental Results. In experiment, the one-dimensional motion trajectory is firstly tested. The step signal with

1/8 Hz square wave and 5 mm amplitude is sent to the x -axis tables as the position command to cause tables motion. The experimental results in position response and current response using FC are shown in Figure 12. In Figure 12(a), the position response with 0.2 second rising time and 2% overshoot and near zero steady state error which shows a good response performance. In Figure 12(b), the current in i_d axis approaches zero which reveals the success of the vector control in PMSM drive. To evaluate the tracking performance

FIGURE 15: (a) Star motion trajectory tracking; (b) tracking error; (c) position response; (d) current response.

in multiaxis motion table, the indices are firstly defined as follows:

$$T(k) = \sqrt{\sum_{i=1}^n (S_{pi}^*(k) - S_{pi}(x))^2}, \quad (25)$$

$$mte = \sum_{k=1}^m \frac{T(k)}{m},$$

where $T(k)$, $S_{pi}^*(k)$, $S_{pi}(k)$, n , m , and mte , respectively, represent instantaneous error, i th-axis position command, i th-axis position feedback, number of motion table, total acquisition data, and mean tracking error. Further, in two-dimensional motion trajectory, the circular motion trajectory control with center (50, 50) mm and radius 25 mm is evaluated and the experimental results in motion trajectory tracking, tracking error, position response, and current response using same FC are shown in Figure 13. The tracking errors in Figure 13(b) show the maximum ± 0.46 mm in x -axis table, maximum

FIGURE 16: (a) Helical circular motion trajectory and (b) tracking results.

FIGURE 17: Motion trajectory tracking and tracking error of STUST character.

± 0.35 mm in y -axis table, and 0.1759 mm mean tracking error in overall circular motion. It presents a good contour tracking result. Additionally, the current response of i_d -axis both in x - and y -axis motors all can be controlled to zero which presents the success of the vector controller. In the window tracking motion, the trajectory is designed as Figure 3(a) and its experimental results in motion trajectory tracking, tracking error, position response, and current response are shown in Figure 14. The tracking errors in Figure 14 show the maximum ± 0.2 mm in x -axis table, maximum ± 0.2 mm in y -axis table, and 0.054 mm mean tracking error in overall motion. In the star tracking motion, the trajectory is designed as Figure 3(b) and its experimental results are shown in Figure 15. The tracking errors in Figure 15 show the maximum ± 0.4 mm in x -axis table, maximum ± 0.55 mm in y -axis table, and 0.0698 mm mean tracking error in overall star motion. However, the larger tracking error in both window and star motion trajectory occurred at the position of command suddenly changed; otherwise, the tracking error is very small. The results show a good contour tracking response in both window and star motion trajectory. Finally, the three-dimensional motion trajectory which is helical circular trajectory and "STUST" character trajectory are evaluated, and their experimental tracking results are shown in Figures 16 and 17, respectively. In Figure 16(b), the tracking errors show that it can keep within ± 0.2 mm in x - and y -axis table motion, ± 0.01 mm in z -axis table motion, and 0.1162 mm mean tracking error in overall helical circular motion which reveals a good tracking response. In performing the motion of "STUST" character, the z -axis table just runs the up and down motion, the "STUST" character is drawn by the simultaneous motion of x - and y -axis table. The tracking error in Figure 17 shows the maximum ± 0.3 mm in x -axis, y -axis, and z -axis table and less than 0.3 mm in overall motion tracking. Therefore, Figures 16–17 show a good tracking result in 3D motion contour tracking. Therefore, Figures 12–17 present that the proposed FPGA-based multiaxis motion control IC is effectiveness and correctness.

6. Conclusions

The FPGA-based multiaxis motion controller has been successfully demonstrated from ModelSim/Simulink cosimulation to FPGA realization experiment in this paper. For a multiaxis motion controller system, DSP and FPGA all can provide a good solution in implementation. However, due to one DSP chip only can support one CPU inside and, 12 channels PWM output and 2 QEP output in its peripheral circuit, it is suitable for a two-axis motion controller system. If we want to realize a multiaxis motion controller system, it needs several DSP chips. In contrast with DSP, FPGA has the properties of fast computing power, parallel processing, and hardware-software codesign which cause the multiaxis motion controller system to be implemented by one FPGA chip. Finally, the work herein is summarized as follows.

- (a) The functionalities required to build a multiaxis servo controller and motion trajectory planning can be integrated in one FPGA chip.

- (b) The computation of key algorithms, such as SVPWM, CCCT, and FC in PMSM drive can be completed within $2\ \mu\text{s}$ by FPGA implementation.
- (c) The VHDL code of the proposed multiaxis servo controller IP has been effectively verified from ModelSim/Simulink cosimulation results.
- (d) Some experimental results by position step response, circular, window, star, and helical circular motion trajectory tracking, the specific character ("STUST") motion trajectory tracking, and so forth have revealed that the proposed FPGA-based multiaxis motion controller is effectiveness and correctness.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

References

- [1] J. U. Cho, Q. N. Le, and J. W. Jeon, "An FPGA-based multiple-axis motion control chip," *IEEE Transactions on Industrial Electronics*, vol. 56, no. 3, pp. 856–870, 2009.
- [2] X. Shao and D. Sun, "Development of a new robot controller architecture with FPGA-based IC design for improved high-speed performance," *IEEE Transactions on Industrial Informatics*, vol. 3, no. 4, pp. 312–321, 2007.
- [3] S. S. Yeh and J. T. Sun, "Design of perfectly matched zero-phase error tracking control for multi-axis motion control systems," in *Proceedings of the SICE Annual Conference*, pp. 528–533, August 2012.
- [4] Y. S. Kung, R. F. Fung, and T. Y. Tai, "Realization of a motion control IC for X-Y table based on novel FPGA technology," *IEEE Transactions on Industrial Electronics*, vol. 56, no. 1, pp. 43–53, 2009.
- [5] S.-S. Yeh and P.-L. Hsu, "Analysis and design of integrated control for multi-axis motion systems," *IEEE Transactions on Control Systems Technology*, vol. 11, no. 3, pp. 375–382, 2003.
- [6] Z. Zhou, T. Li, T. Takahashi, and E. Ho, "FPGA realization of a high-performance servo controller for PMSM," in *Proceedings of the 19th Annual IEEE Applied Power Electronics Conference and Exposition (APEC '04)*, vol. 3, pp. 1604–1609, Anaheim, Calif, USA, February 2004.
- [7] B. K. Bose, *Power Electronics and AC Drives*, Prentice Hall, 1986.
- [8] V. Wilson, P. Agarwal, and S. P. Srivastava, "Performance investigation of DSP based self-controlled PMSM drive," in *Proceedings of the Annual IEEE India Conference (INDICON '12)*, pp. 238–243, Kochi, India, December 2012.
- [9] E. Monmasson, L. Idkhajine, M. N. Cirstea, I. Bahri, A. Tisan, and M. W. Naouar, "FPGAs in industrial control applications," *IEEE Transactions on Industrial Informatics*, vol. 7, no. 2, pp. 224–243, 2011.
- [10] E. Monmasson, L. Idkhajine, and M. W. Naouar, "FPGA-based controllers," *IEEE Industrial Electronics Magazine*, vol. 5, no. 1, pp. 14–26, 2011.
- [11] Y. S. Kung, V. Q. Nguyen, T. H. Nguyen, C.-C. Huang, and L.-C. Huang, "Simulink/modelsim co-simulation and FPGA realization of speed control IC for PMSM drive," *Procedia Engineering*, vol. 23, pp. 718–727, 2011.

- [12] L. Idkhajine, E. Monmasson, M. W. Naouar, A. Prata, and K. Bouallaga, "Fully integrated FPGA-based controller for synchronous motor drive," *IEEE Transactions on Industrial Electronics*, vol. 56, no. 10, pp. 4006–4017, 2009.
- [13] B. Alecsa, M. N. Cirstea, and A. Onea, "Simulink modeling and design of an efficient hardware-constrained FPGA-based PMSM speed controller," *IEEE Transactions on Industrial Informatics*, vol. 8, no. 3, pp. 554–562, 2012.
- [14] H.-H. Chou, Y.-S. Kung, N. Q. Quynh, and S. Cheng, "Optimized FPGA design, verification and implementation of a neuro-fuzzy controller for PMSM drives," *Mathematics and Computers in Simulation*, vol. 90, pp. 28–44, 2013.
- [15] Y. Li, J. Huo, X. Li, J. Wen, Y. Wang, and B. Shan, "An open-loop sin microstepping driver based on FPGA and the co-simulation of modelsim and simulink," in *Proceedings of the International Conference on Computer, Mechatronics, Control and Electronic Engineering (CMCE '10)*, pp. 223–227, August 2010.
- [16] The Mathworks, "Matlab/Simulink Users Guide, Application Program Interface Guide," 2004.
- [17] Modeltech, *ModelSim Reference Manual*, 2004.
- [18] F.-J. Lin, P.-H. Shieh, and P.-H. Shen, "Robust recurrent-neural-network sliding-mode control for the X-Y table of a CNC machine," *IEE Proceedings: Control Theory and Applications*, vol. 153, no. 1, pp. 111–123, 2006.