

Article

FPGA-Based Hybrid Stepper Motor Drive System Design by Variable Structure Control

Chiu-Keng Lai *, Bo-Wei Lin, Hsiang-Yueh Lai and Guan-You Chen

Department of Electrical Engineering, National Chin-Yi University of Technology, Taiping Dist., Taichung 411030, Taiwan; s4a712102@gm.ncut.edu.tw (B.-W.L.); anne@ncut.edu.tw (H.-Y.L.); s4a712105@gm.ncut.edu.tw (G.-Y.C.)

* Correspondence: chiukl@ncut.edu.tw; Tel.: +886-4-2392-4505 (ext. 7216); Fax: +886-4-2392-4419

Abstract: A stepper motor is usually operated in position open-loop control for simplicity. However, in order to improve the transient and steady-state performances of the stepper motor-based drive system, a high performance stepper motor drive system is usually designed to feed the encoder signal back to form a closed-loop system such as a general servo motor drive, and high-performance position and speed loop controller can then be used to overcome the nonlinear characteristics of the motor, such as the cogging force and impacts from external load. On the other hand, the existed cogging force of the hybrid stepper motor must be solved to increase the positioning precision. The variable structure control (VSC) is insensitive to the bounded uncertainty and load disturbance, and has been known as a high-performance controller. A predefined sliding surface is used to shape the system performances, and incorporate with switching control to achieve the robustness property. Thus, we applied the VSC to implement the stepper motor drive system about the position and speed control, and the switching control is used to overcome the cogging force. The field programmable gate array (FPGA) is a good alternative to be used to realize a motor drive system by considering its programmable ability and diverse designing environment. It is easily developed as an intellectual property (IP) for future use or combined as a part of a large control system. This paper showed the procedures to develop the hardware circuits for the variable structure controller, and applied it to stepper motor position and velocity control. Functions such as PI controllers, dq-axis transformation and two-phase space vector modulation (SVPWM) designed for the stepper motor drives are also shown in the paper. The system simulation and hardware circuit realization are based on MATLAB/Simulink, and realized on Altera FPGA. Simulations on MATLAB/Simulink with trapezoidal velocity profile command, and experiments with and without the load added are shown to demonstrate the hardware performances and correctness.

Keywords: hybrid stepper motor; variable structure control; FPGA; position control; speed control



Citation: Lai, C.-K.; Lin, B.-W.; Lai, H.-Y.; Chen, G.-Y. FPGA-Based Hybrid Stepper Motor Drive System Design by Variable Structure Control. *Actuators* **2021**, *10*, 113. <https://doi.org/10.3390/act10060113>

Academic Editor: Gianluca Rizzello

Received: 18 April 2021

Accepted: 25 May 2021

Published: 28 May 2021

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2021 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Considering the advantages of low cost, high precision, and high torque output, stepper motors are frequently used in the field of high-precision control. Traditional open-loop control can lead to momentary step loss or position error when the load changes. As a result, the stepper motor drive system for position and speed control loops has been gradually enhanced as a closed-loop control [1–5]. In Ref. [1], PI controllers are applied for the current and speed closed-loop control, and P control for the position control loop. For Ref. [2], the closed-loop control structure and PI controllers with motor parameter identification are applied to the current and position control of a two-phase bipolar hybrid stepper motor. Furthermore, feedforward compensation is used for the damping control such that the motor can be operated at a high speed range. The authors of Ref. [3] adopted the fuzzy-sliding mode observer to estimate the motor velocity and fed it back as a speed closed-loop system. In that system, PI controllers are used for the stationary A–B phase current control. In Ref. [4], the authors proposed a wide speed range closed-loop-control

drive system, and implemented by a field programmable gate array (FPGA) chip. For low-speed operation, the speed control loop is bypassed, and for the high-speed operation, the field-weaken vector control is used to reach the desired performance. Regarding Ref. [5], it proposed a simple torque control method with shaft-position feedback to correct the load angle, and was also realized by FPGA. In Ref. [6], the PID controller and feedforward control are applied to the stepper motor current control to guarantee the desired current response.

For most of the AC motor drive systems, such as induction motor (IM) and permanent magnet synchronous motor (PMSM)-based systems, vector control is widely used. For such systems, a three-phase modulator is usually used to generate the desired d-axis and q-axis voltages. Clarke and inverse Clarke matrices are used for the transformation between the three-phase and two-phase systems. For a two-phase hybrid stepper motor, it only needs a two-axis modulator mainly made by two H-bridge inverters to convert the DC bus voltage into a pair of quadrature voltages for the two stator coils. Thus, the stepper motor drive system can be operated without executing the conversion of a three-phase stationary (a-b-c) coordinate system into two-phase stationary (α, β) coordinate system. Regarding the vector control for the stepper motor drive system, for the implementation of the vector control in the stepper motor, only Park and inverse Park matrices are required.

The cogging force of the hybrid stepper motor affects the accuracy of positioning. Moreover, for the vector control protocol, the coordinate transformation requires the position feedback to convert the stationary $\alpha\beta$ -axis signals into synchronous rotary dq-axis signals. For a closed-loop stepper motor drive system, a high-precision incremental encoder can be used to improve the positioning accuracy. However, the coordinate transformation performs the sin/cos calculation. This requires the position feedback, and the resolution of the sin/cos table holds the precision of the transformation. Additionally, a high-resolution position feedback by incremental encoder is hard to produce the same resolution of voltage generation because of the precision limitations of the space vector pulse width modulation (SVPWM) hardware system, and that leads the current control to not be as precise as the position control loop. As a result, a high-performance variable structure controller is used to improve the positioning accuracy regarding the existence of the problems mentioned above. The variable structure control (VSC) has the advantages of insensitivity to parameter changes, external interference suppression and fast dynamic response [7]. The VSC or sliding mode control (SMC) has been widely applied to single-motor drive design [8–10], or designed for a system such as Ref. [11] which is applied to the balance control of a two-wheel vehicle system. For systems controlled by VSC, some steps are included into the design procedure. First, a mathematic model for the control system is presented. Second, a sliding surface or switching surface which dynamics is equivalent to the desired system performances is made. Third, a control law to guarantee the existence of the sliding mode is made. Sometimes, a VSC system exists in the reaching phase and sliding phase, and the system performance is guaranteed on its sliding phase but not on its reaching phase [7,8]. Furthermore, finding an easy way to shape the desired system dynamics is also important for VSC systems. Thus, for the proposed algorithm, the design procedure of a stepper motor drive system is demonstrated step by step. The controlled system is first represented into state space form, and the linear state feedback control method is applied to find the desired system dynamics and switching surface. Finally, a control law which combines the linear control and VSC is designed to make the sliding condition occur.

Hardware circuits realized by field programmable gate array (FPGA) have the benefit of fast prototyping with digital structure, verified with ease. In addition to associating with some of the high-level programming languages, such as C/C++ or MATLAB/Simulink, control systems designed and represented by hardware description language (HDL) are easily achieved. Furthermore, owing to the development of high-density and high-performance FPGA [12–14], hardware controllers implemented by FPGA are widely obtained. Some examples include the design of velocity estimation [15], virtual anemometer [16], PID controller [17] and fuzzy controller [18]. FPGA is also suitable for the control system realization of power electronic circuits [19]. In this paper, we first constructed the mathematical model

of a hybrid stepper motor and its drive control system in Simulink [4,9]. Next, the models of the controllers were then digitized to be converted into Verilog HDL codes to meet the requirements of hardware design resources. Finally, the designed hardware circuit was used to implement the position, speed and current control to illustrate the correctness of the hardware circuit design and the performance of the variable structure controller. The performances are mainly demonstrated by a 4-ch digital oscilloscope with logical input. The main contributions of the proposed model are based on the state space motor model to the position and speed control, and the applied variable structure controller reaching the desired position and speed performance. The current loop is based on a two-phase vector control strategy where the d- and q-axis loops are controlled separately. The designed system was successfully implemented on FPGA and evaluated by hybrid stepper motor control platform.

The further contents of the paper are as follows: in Section 2, the modeling of the hybrid stepper motor is shown, and the variable structure controller for position/speed control is introduced. Section 3 gives the position variable structure controller design. The simulations on MATLAB/Simulink platform of the proposed are given in Section 4. The experimental setup and results are shown in Section 5. Finally, Section 6 contains the conclusions.

2. The Modeling of Two-Phase Hybrid Stepper Motor and Variable Structure Control

2.1. Mathematic Model of Two-Axis Hybrid Stepper Motor

The electrical circuit equivalent for the two-phase hybrid stepper motor is:

$$v_a = Ri_a + L \frac{di_a}{dt} - K_e \omega_m \sin(N_r \theta_m) \quad (1)$$

$$v_b = Ri_b + L \frac{di_b}{dt} + K_e \omega_m \cos(N_r \theta_m) \quad (2)$$

where v_a is the voltage of the coil a, v_b is the voltage of the coil b, i_a is the current of the coil a, i_b is the current of the coil b, R is the coil resistance, L is the coil inductance, K_e is the motor back-EMF constant, ω_m is the mechanical speed of the motor, N_r is the number of rotor teeth, and θ_m is the rotor position. The equivalent circuit of the two-phase hybrid stepper motor is shown in Figure 1.

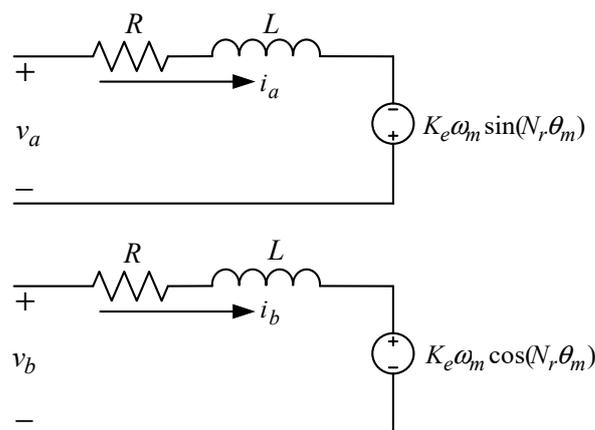


Figure 1. The equivalent stator circuit of two-phase hybrid stepper motor.

The generated torque output without considering the component of cogging torque is:

$$\begin{aligned} T_e &= K_m [-i_a \sin(N_r \theta_m) + i_b \cos(N_r \theta_m)] - F_c \sin(4N_r \theta_m) \\ &= T'_e - F_c \sin(4N_r \theta_m) \end{aligned} \quad (3)$$

where K_m is the motor torque constant, F_c is the cogging force constant, and T'_e is defined as:

$$T'_e = K_m[-i_a \sin(N_r \theta_m) + i_b \cos(N_r \theta_m)], \tag{4}$$

which is the generated torque by the stator coil currents, i_a and i_b . The mechanical dynamic equation is:

$$\begin{aligned} \frac{d\omega_m}{dt} &= \frac{1}{J_m}(-B_m \omega_m - T_L + T'_e - F_c \sin(4N_r \theta_m)) \\ &= \frac{1}{J_m}(-B_m \omega_m - T'_L + T'_e) \end{aligned} \tag{5}$$

where J_m is the inertia of the motor, B_m is the viscous friction coefficient of the motor, T_L is the load torque, and $T'_L = T_L + F_c \sin(4N_r \theta_m)$ is the lumped total load for the dynamical control system.

To construct the mathematical model of the hybrid stepper motor with MATLAB/Simulink, Equations (1) and (2) can be rewritten as:

$$\frac{d}{dt} i_a = -\frac{R}{L} i_a + \frac{K_e}{L} \omega_m \sin(N_r \theta_m) + \frac{v_a}{L} \tag{6}$$

$$\frac{d}{dt} i_b = -\frac{R}{L} i_b - \frac{K_e}{L} \omega_m \cos(N_r \theta_m) + \frac{v_b}{L} \tag{7}$$

From Equations (4)–(7), the built models by MATLAB/Simulink are shown in Figure 2, where the blocks “Ia” and “Ib” are equivalent to Equations (6) and (7); cos_sin block is used to generate the sine and cosine functions, and the blocks “Te” and “Wm” are used to realize Equations (4) and (5).

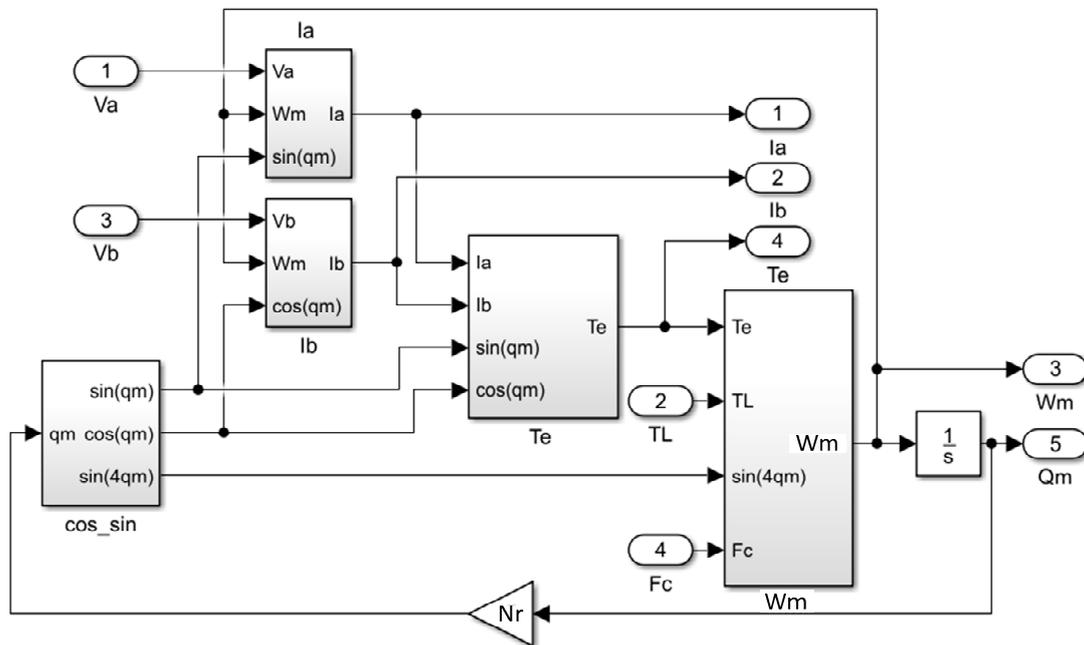


Figure 2. The complete stepper motor module.

2.2. Variable Structure Control

In the following, the variable structure controller will be used as the main controller for the position and speed control.

A single-input linear system under nominal condition is expressed in a controlled canonical form as:

$$\dot{x} = Ax + bu, \tag{8}$$

where x is the system state, u is the scalar control input, and A and b are with the appropriate range matrices. The control law u by variable structure system can be divided

into two parts: the linear part u_L and the VSC part u_{VSC} . First, let the nominal system be under linear state feedback control; we can easily have the control gain, \mathbf{k}^T to complete the feedback control.

$$u_L = -\mathbf{k}^T \mathbf{x}, \quad \mathbf{k}^T = [k_1, k_2, \dots, k_n] \quad (9)$$

Some techniques such as the pole placement or linear quadratic method can be used to determine the value of \mathbf{k} . Under the condition of a closed loop control, from [7], (8) can be expressed as:

$$\dot{\mathbf{x}} = [\mathbf{A} - \mathbf{b}\mathbf{k}^T] \mathbf{x} = \mathbf{A}_c \mathbf{x} \quad (10)$$

Next, the sliding surface of the variable structure controller is defined as:

$$\sigma(\mathbf{x}, t) = \mathbf{c}^T [\mathbf{x} - \mathbf{x}_0] - \mathbf{c}^T \mathbf{A}_c \int_0^t \mathbf{x}(\tau) d\tau, \quad (11)$$

and \mathbf{c} is a constant vector.

For the closed-loop system (10) under nominal conditions, for any initial value \mathbf{x}_0 , $\sigma(\mathbf{x}, t) = 0$ for $t \geq 0$; i.e., the system states are initially on the sliding surface and will stay on it for all the next proceeding time. However, for a perturbed system, the linear control element (9) may not make the sliding condition exist. We require additional control to make and keep the states on the sliding surface. The additional control law is defined as:

$$u_{VSC} = -q \operatorname{sgn}(\sigma) \quad (12)$$

where $\operatorname{sgn}(\cdot)$ is the sign function, and q is a positive scalar constant. Then, the combined control law for the designed system is:

$$u = u_L + u_{VSC} = -\mathbf{k}^T \mathbf{x} - q \operatorname{sgn}(\sigma) \quad (13)$$

To determine the quantity of q , it must be chosen to satisfy the sliding condition,

$$\sigma \dot{\sigma} < 0 \quad (14)$$

Thus, the desired closed-loop dynamics for the perturbed system can be obtained even though the perturbations exist.

3. Position Control by State Feedback and Variable Structure Control

The vector control methodology is used for the motor drive system design, the Park transformations between the stationary reference frame to synchronous rotation reference frame for winding voltages and currents are, respectively, defined by:

$$\begin{bmatrix} v_d \\ v_q \end{bmatrix} = \begin{bmatrix} \cos(N_r \theta) & \sin(N_r \theta) \\ -\sin(N_r \theta) & \cos(N_r \theta) \end{bmatrix} \begin{bmatrix} v_a \\ v_b \end{bmatrix} \quad (15)$$

and

$$\begin{bmatrix} i_d \\ i_q \end{bmatrix} = \begin{bmatrix} \cos(N_r \theta) & \sin(N_r \theta) \\ -\sin(N_r \theta) & \cos(N_r \theta) \end{bmatrix} \begin{bmatrix} i_a \\ i_b \end{bmatrix} \quad (16)$$

where v_d, v_q, i_d and i_q are direct voltage, quadrature voltage, direct current, and quadrature current in synchronous rotation reference frame. Applying Park transformation to the state-space equation yields new state-space equation as follows:

$$\frac{d}{dt} i_d = \frac{1}{L} (v_d - R i_d + N_r L \omega_m i_q) \quad (17)$$

$$\frac{d}{dt} i_q = \frac{1}{L} (v_q - R i_q - N_r L \omega_m i_d - K_e \omega_m) \quad (18)$$

With the vector control assumed, the instant motor generated torque by stator coil current can be expressed as:

$$T_e' = K_m i_q \quad (19)$$

Applying the state feedback position controller design, the system model is first described in the state variable form, as shown in (20):

$$\begin{bmatrix} \dot{\theta}_m \\ \dot{\omega}_m \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 0 & -B_m/J_m \end{bmatrix} \begin{bmatrix} \theta_m \\ \omega_m \end{bmatrix} + \begin{bmatrix} 0 \\ 1/J_m \end{bmatrix} (T_e' - T_L') \quad (20)$$

For a desired rotor position θ_d , we first define two new variables, z_1 and z_2 as:

$$\begin{cases} z_1 = \theta_m - \theta_d \\ z_2 = \omega_m \end{cases}, \quad (21)$$

where z_1 is the position error, and z_2 is its derivative. Substituting (19) and (21) into (20), the dynamic error equation for position control can be obtained as (22):

$$\begin{bmatrix} \dot{z}_1 \\ \dot{z}_2 \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 0 & -a \end{bmatrix} \begin{bmatrix} z_1 \\ z_2 \end{bmatrix} + \begin{bmatrix} 0 \\ b \end{bmatrix} i_q - \begin{bmatrix} 0 \\ 1/J_m \end{bmatrix} T_L' \quad (22)$$

where $a = B_m/J_m$ and $b = K_m/J_m$, the matrix \mathbf{A} and vector \mathbf{b} of (8) are:

$$\mathbf{A} = \begin{bmatrix} 0 & 1 \\ 0 & -a \end{bmatrix}, \quad \mathbf{b} = \begin{bmatrix} 0 \\ b \end{bmatrix} \quad (23)$$

The pole placement method is used to design the position control system characteristics. At first, we do not have to consider the uncertainty and interference. The feedback gain \mathbf{k} of (9) is selected for the nominal condition system. The desired system characteristic equation is shown in (24),

$$s^2 + (a + bk_2)s + bk_1 = 0 \quad (24)$$

where a and b are defined in (23), in which case the poles of the system are located at:

$$s_{1,2} = \frac{-(a + bk_2) \pm \sqrt{(a + bk_2)^2 - 4bk_1}}{2} \quad (25)$$

At first, by appropriately selecting k_1 and k_2 in (24) to meet the requirements, such as the damping ratio, ζ , and rise time, t_r [20]. Second, the sliding surface can be yielded by Equation (11). The matrix \mathbf{A}_c in (11) is the equivalent system matrix under linear state feedback control gain \mathbf{k} . The constant vector \mathbf{c}^T can be simply determined by choosing $\mathbf{c}^T \mathbf{b} = 1$. Since the matrix \mathbf{b} of the two-phase hybrid stepper motor system is $\mathbf{b}^T = [0 \quad b]$, \mathbf{c} can thus be set as:

$$\mathbf{c}^T = [0 \quad 1/b] \quad (26)$$

The VSC compensation part, $-q \operatorname{sgn}(\sigma)$, is used to overcome the parameters' uncertainties and external interferences, including the load and cogging torque. For a motor position control system, the form of the total disturbance p is $\mathbf{p}^T = [0 \quad p]$, so the value of q should be set to satisfy the requirement:

$$q > |p|_{\max} \quad (27)$$

where $|p|_{\max}$ is the upper bound of total perturbation. Note that, with the constraint of (27), the designed system is stable and robust to a bounded uncertainty and load.

4. Computer Simulation and Results

Simulations are based on MATLAB/Simulink platform to verify the system performance. The parameters of the two-phase hybrid stepper motor used in the simula-

tion are listed in Table 1. Two kinds of command are assumed: (a) position step input, (b) motion control with trapezoidal speed profile limitation. The step input command is 0.628 radian. The motion command includes the acceleration segment, deceleration segment and constant speed, and total angular displacement 0.628 radian in the duration time 0.7 s. Substituting the parameters shown in Table 1 into (22), the control gain $\mathbf{k}^T = [k_1 \ k_2] = [15 \ 0.4]$ is chosen such that the resulting second-order system (23) possesses the performances of damping ratio 0.707 and rising time 0.05 s. Thus, the linear control law u_L and sliding surface σ are determined as:

$$u_L = -\mathbf{k}^T \mathbf{x} = -[15 \ 0.4] \begin{bmatrix} z_1 \\ z_2 \end{bmatrix} \tag{28}$$

and

$$\sigma[\mathbf{x}(z_1, z_2), t] = 0.000273 z_2 + \int_0^t (15z_1 + 0.4z_2) d\tau \tag{29}$$

Table 1. The parameters of the simulated motor.

Rated Voltage	3.6 V
Rated current	2.0 A
Coil resistance (R)	1.6 Ω
Coil inductance (L)	0.0063 H
Rotor inertia (J_m)	$4.93 \times 10^{-5} \text{ N} \cdot \text{m}/\text{s}^2$
Coefficient of friction (B_m)	$4.93 \times 10^{-6} \text{ N} \cdot \text{m}/\text{s}$
Torque constant (K_m)	0.18 N · m/A
Back EMF (K_e)	0.18 V/rad · s ⁻¹
Cogging torque constant (F_c)	0.006 N · m

According to (28) and (29), and with the setting,

$$q = q_0 + q_1 |\omega_m|, \tag{30}$$

where q_0 and q_1 are two positive constants for the VSC control part. The schematic block diagram of the Simulink model of the variable structure controller and the structure of the drive system are shown in Figure 3.

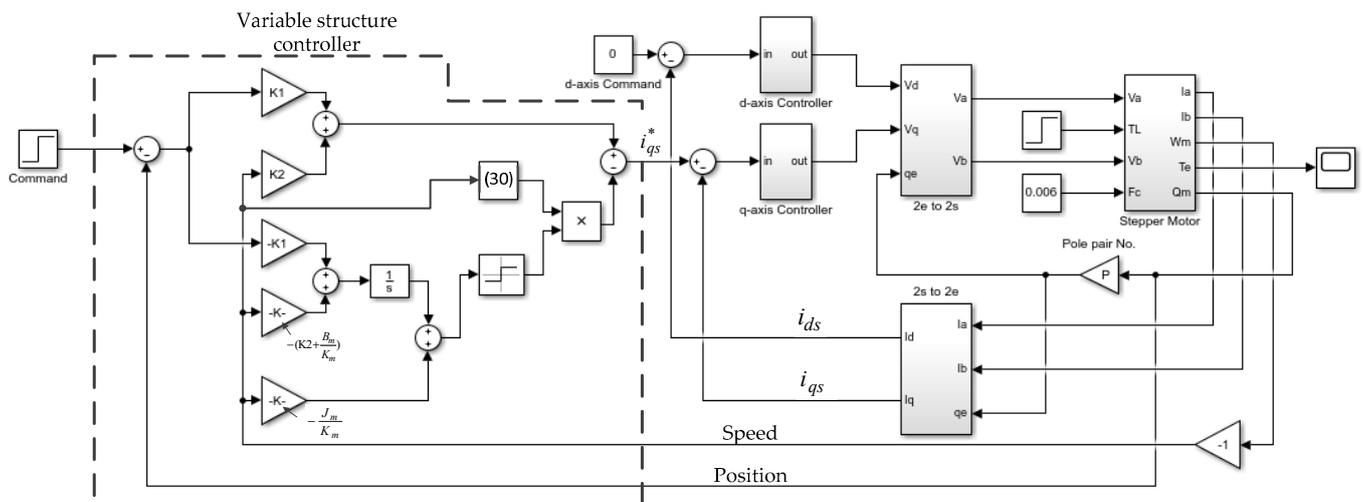


Figure 3. The schematic block diagram of VSC-based stepper motor drive system.

The system shown in Figure 3 includes the linear state feedback and variable structure control for the position/speed loops, and the inner d- and q-axis current control loops are

dominated by two PI controllers. As shown in Figure 3, the command of q-axis current control loop is from the output of the variable structure controller, and the command of the d-axis current control loop is set as 0. The sampling time for the current and speed/position control loops are set as 0.05 ms and 0.5 ms, respectively.

A typical trapezoidal speed profile with a displacement 0.628 radian, as shown in Figure 4, is used as the command input. The following simulated responses, including the position, speed, current and sliding surface σ , will be shown for with and without load conditions, which are defined dependent on whether or not the disk load axis connected. The speed profile of Figure 4 is made with the time constraints of acceleration period 0.2 s, constant speed period 0.3 s and deceleration period 0.2 s, and the variable structure control part (29) is set as:

$$q = q_0 + q_1|\omega_m| = 0.05 + 0.1|\omega_m| \quad (31)$$

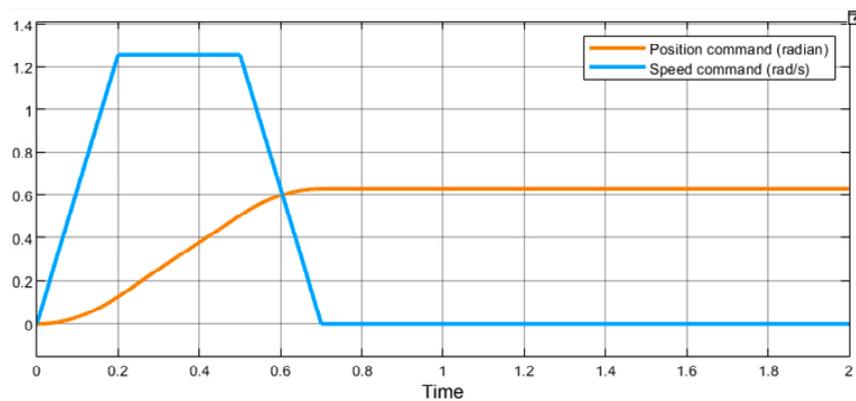


Figure 4. The position command and speed command profiles.

In (31), q_0 is mainly used to reduce the steady-state error coming from the dead-band, backlash, cogging torque, or other nonlinear part of the system, and make the sliding condition exist. The second part of (31), whereby magnitude is proportional to the motor speed, is designed to guarantee the system states follow the desired trajectory defined in matrix A_c .

First, the responses of step position command input are shown in Figure 5. There is no speed restriction on the response, and the speed and position are controlled by the equivalent system matrix A_c . Figures 6–8 are the position, speed and current response under the no-load condition. In Figure 6, we can see that the feedback has reached the target position of 0.628 radians at the time of about 0.7 s. The simulation shows that the drive system with variable structure control can ensure the performance of the position control. The speed response displayed on Figure 7 shows that the acceleration segment, constant speed and deceleration are near to the desired speed command profile. From Figure 7, the speed at the run segment is about 1.256 rad/s. Due to the chatter control of VSC and the low-speed operation, there is a little ripple in the speed response. The current response is shown in Figure 8, in which the magnitude is small because a no-load condition is asserted.

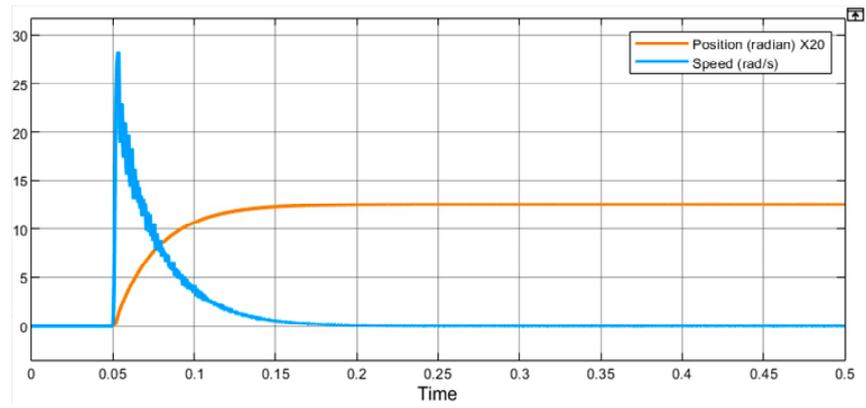


Figure 5. The position and speed response with step input command.

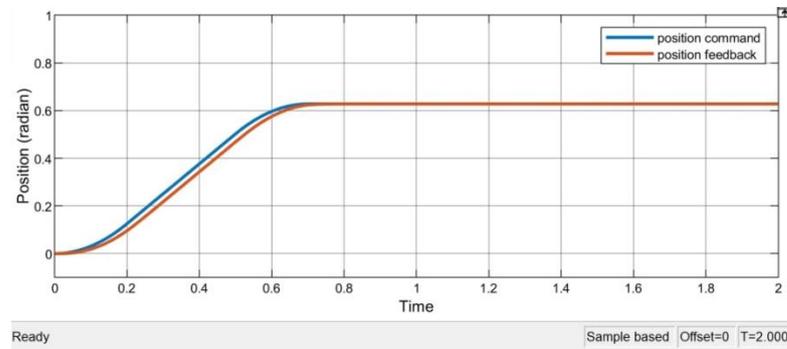


Figure 6. The position command and position response without load.

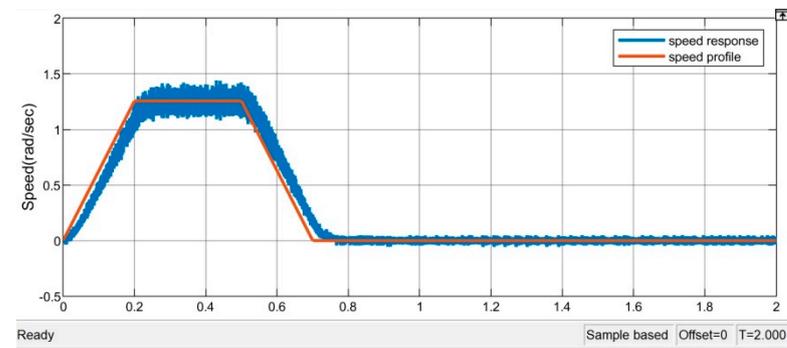


Figure 7. The speed response without load and speed command profile.

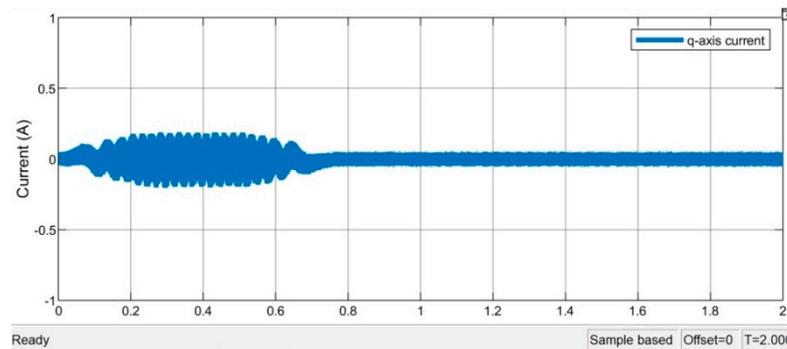


Figure 8. The current response without load.

Figure 9 is the response of the sliding surface σ , which shows that the system enters the sliding surface once the motor starts, and it can be used to verify the correctness of the proposed variable structure control. Sometimes, the motion control is shown running forward and backward, as Figure 10 displays. In it, the position tracking error is defined as the difference between position command and position feedback. Next, the simulation under the load condition is shown; the command used is the same as the no-load condition. For this condition, the inertia and the friction of the rotor are assumed by 10 times and twice the origin, respectively. The simulated results are shown in Figures 11–13.

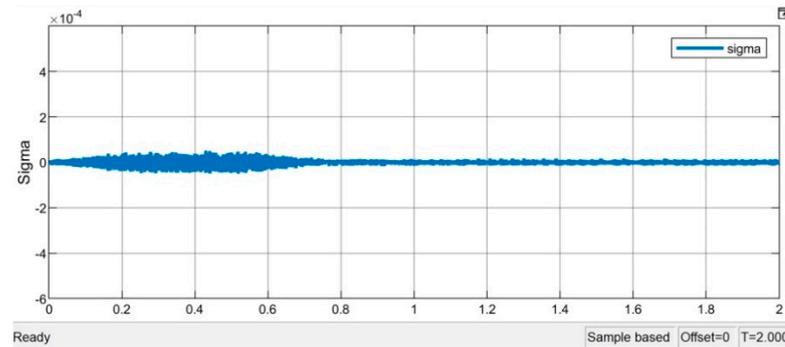


Figure 9. The response of sliding surface σ without load.

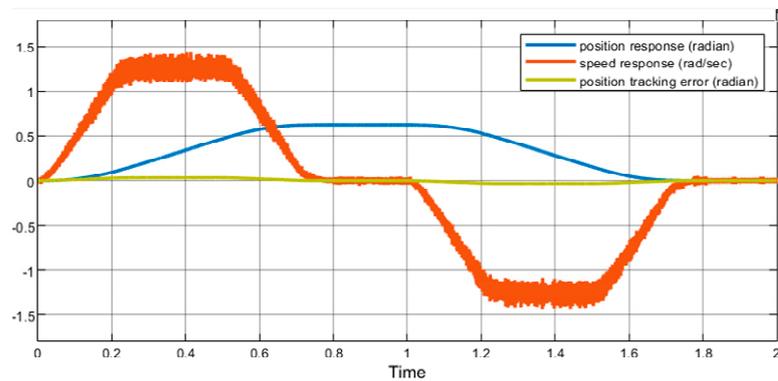


Figure 10. The simulated position, speed and position tracking error.

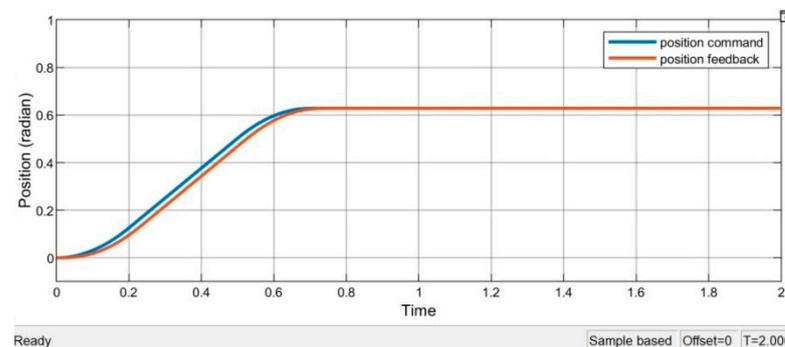


Figure 11. The position response with load.

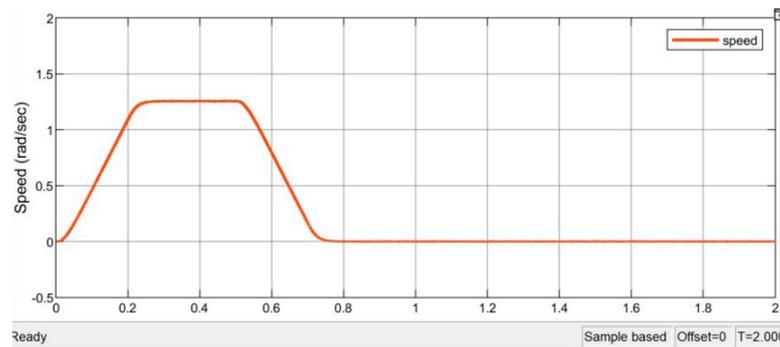


Figure 12. The speed response with load.

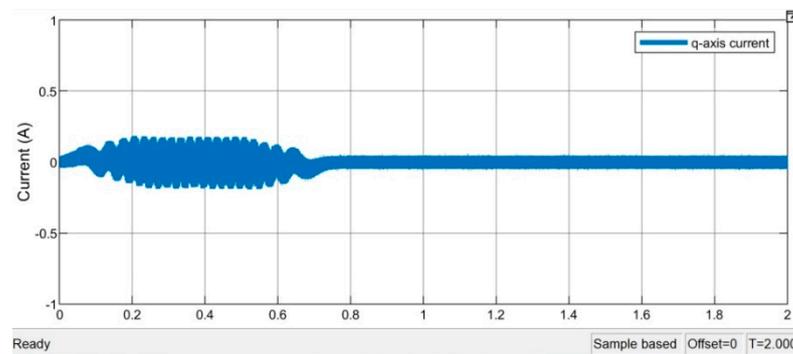


Figure 13. The current response with load.

Figures 11–13 are the position, speed and current responses with load. The position response in Figure 11 is almost the same as the response without load of Figure 6; that is what the VSC control wants. However, because the connected flywheel load makes the system with 10 times the inertia as compared to the unload condition, which reduces the system bandwidth and presents small speed ripples. Since the control law of VSC part is used to make sure the system states keep the sliding condition, it is looking forward to a larger q-axis current for load condition. However, since the system has the same setting for both the unload and load conditions, the current response is nearly the same for both cases, as Figures 8 and 13 have shown. Next, Figure 14 displays the response of σ with load. As the speed ripple of Figure 12 is smaller than Figure 7, the ripple of σ in Figure 14 is smaller than Figure 9, and the system also stays on the sliding phase from the beginning.

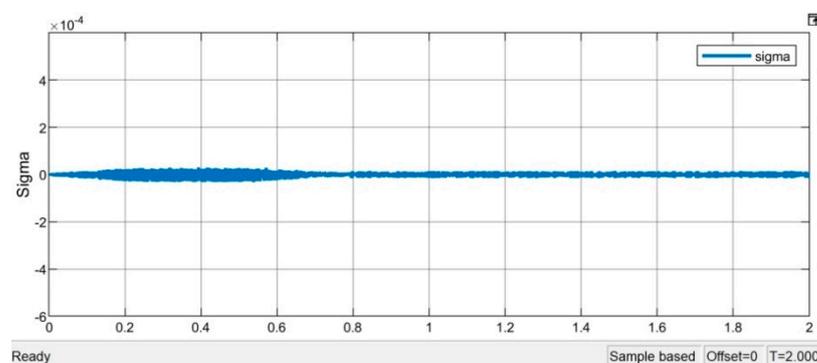


Figure 14. The response of sliding surface σ with load.

5. Experimental Setup and Results

To practically evaluate the actual performance of the proposed control scheme, the experimental setup is shown in Figure 15a, which includes the FPGA board, power board, DC

power supply, hybrid stepper motor, motor load and digital oscilloscope with logic analyzer. On the FPGA board, it includes the Altera Cyclone III FPGA with type EP3C16E144C8N, 12-bit serial analog to digital converter (ADC) AD7866 made by Analog Devices and 12-bit digital to analog converter (DAC) TLV5618 made by Texas Instruments. In the power board, there are two H-bridge circuits used to transfer the DC power to the stepper motor, and two Hall current sensors are set to detect the motor stator currents. The EP3C16E144C8N chip has 84 I/O pins, 15,408 logic elements, 516,096 bits RAM, 56 18×18 multipliers and 4 PLLs [21], and is clocked by 20 MHz. The FPGA chip is used to develop the hardware control system, which includes the coordinate transformation, variable structure controller, PI controllers, 4X quadrature encoder counter, two-phase SVPWM, and the series/parallel conversions for the ADC and DAC converters. A non-optimized usage report about the proposed drive system is shown in Table 2. ADC and DAC are serial type 12-bit converters clocked by 20 MHz. ADC is used to obtain the analog current of phases A and B detected by Hall sensors, and works with a 20 kHz trigger signal. DAC is used to output the corresponding analog voltage signals of the controlled variables such as the rotor position, speed, and sliding surface σ to be checked and compared, and those signals are shown on the digital oscilloscope. DAC is worked with a 2 kHz trigger signal since it is used to demonstrate the mechanic signals. The sampling times for the current control loop and position/speed control loops are, respectively, 0.05 ms and 0.5 ms. The built functions of the FPGA chip and the signal flow diagram out of the FPGA of the experimental system are shown in Figure 16, and signal flow of the blocks inside the FPGA can be found in Figure 3. Logic analyzer, which is built in the digital oscilloscope, is used to capture and display the digital signal outputs from the FPGA. In here, the logical analyzer is used to display the position tracking error of the position control loop. The stepper motor is a hybrid one with 50 rotor teeth, rated current 2 A, and is powered by DC bus voltage 24 V. Additionally, it is equipped with a 5000 p/rev incremental encoder for the position/speed feedback. The load system for the experiment is shown in Figure 15b; it is a flywheel-type load, and counterweights can be used to change the inertia on the rotor. For the comparison purpose, the desired speed waveform and angular destination of the experimental system are the same as the simulated system.

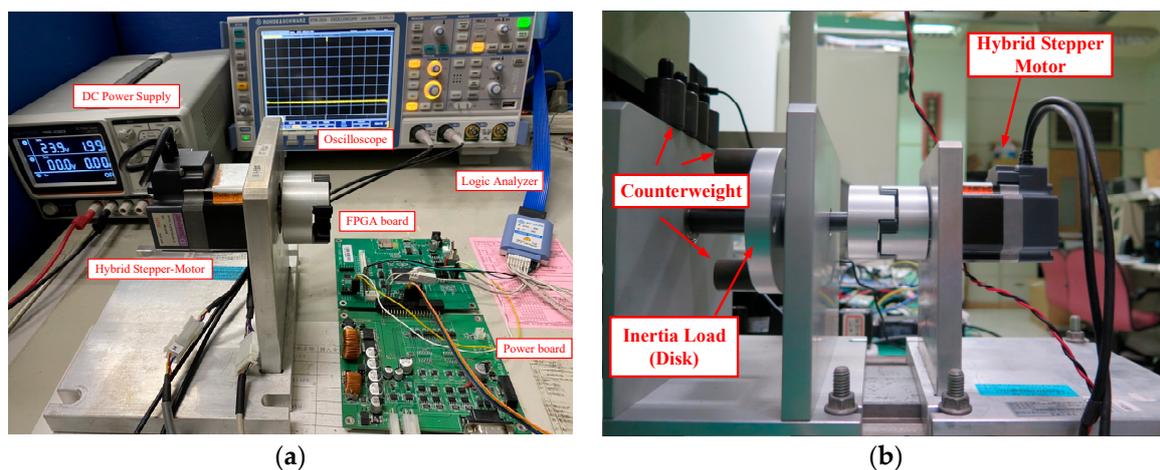


Figure 15. The setup and load for experiment. (a) The FPGA board, power board and stepper motor; (b) stepper motor and flywheel load.

Table 2. The usage of chip to realize the drive system.

Family	Cyclone III
Device	EP3C16E144C8
Total logic elements	5481/15,408 (36%)
Total combinational functions	5153/15,408 (33%)
Dedicated logic registers	1881/15,408 (12%)
Total registers	1881
Total pins	42/85 (49%)
Total virtual pins	0
Total memory bits	155,648/516,096 (30%)
Embedded Multiplier 9-bit elements	64/112 (57%)
Total PLLS	0/4 (0%)

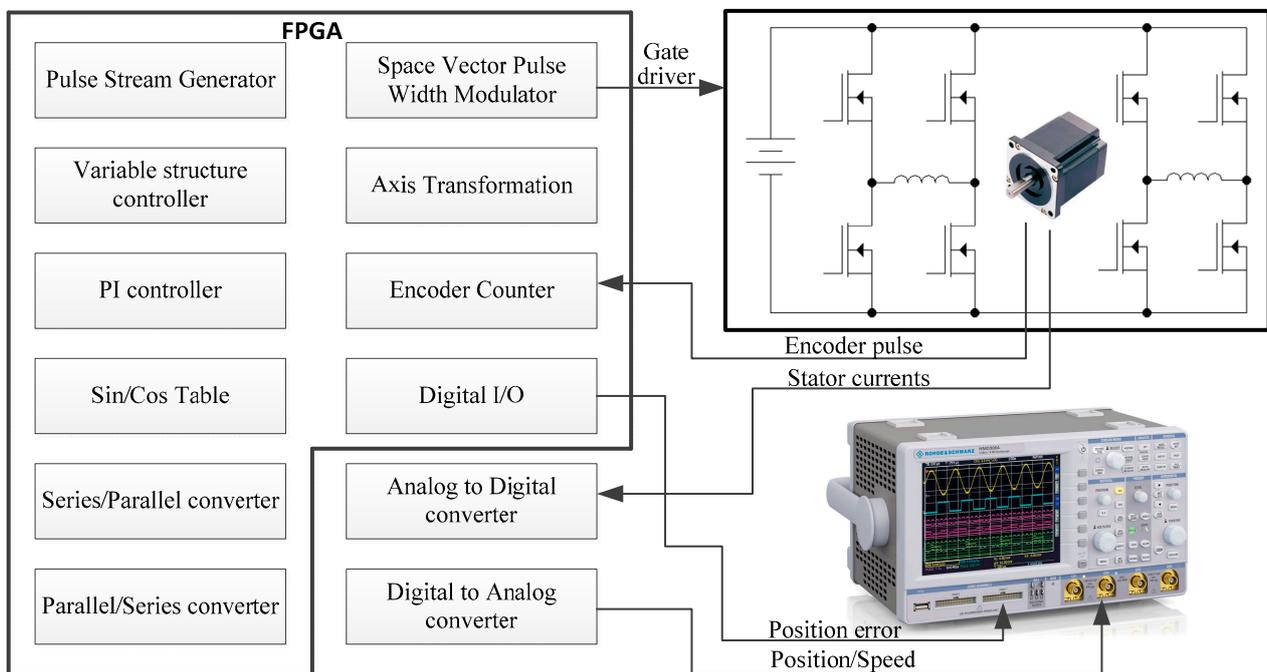


Figure 16. The built functions of FPGA and signal flow diagram of experiment system.

The experimental results with step position command input are shown in Figure 17. The system takes about 100 ms to reach its steady-state value as simulated, and the speed response is not so sharp as the simulation.

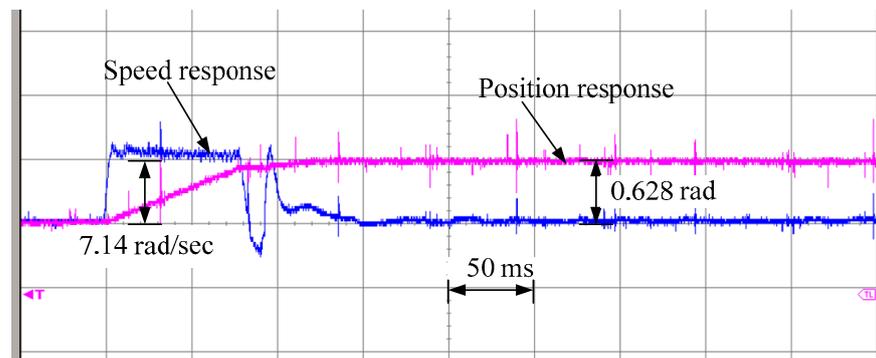


Figure 17. The experimental responses of position and speed response with step command input.

Figure 18 gives the position and speed responses with the 0.628 radian position command. The time base of Figure 18 is 200 ms/DIV. According to the description about the pulse generator, the amounts of total generated pulse are 2000 within the duration [0 s, 0.7 s]. For the position and speed responses, they fit the requirements about the position and speed constraints. To check the position response precisely, we sent the position error in digital format with 8 bits to the output port of the FPGA, and captured them by the logic analyzer. In Figure 18, it displays that the bit D0 is switching between 0 and 1, and the other bits, D1-D7, stay at 0. Thus, the position error in the steady-state condition is about one encoder pulse. Figure 19 is the zoomed plot to the steady-state condition. Figure 20 is the result with running forward and backward.

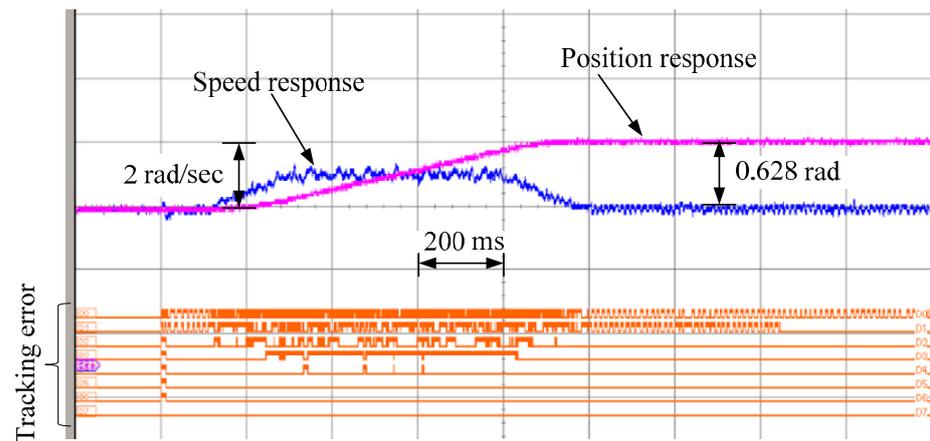


Figure 18. The experimental responses of position and speed without load.

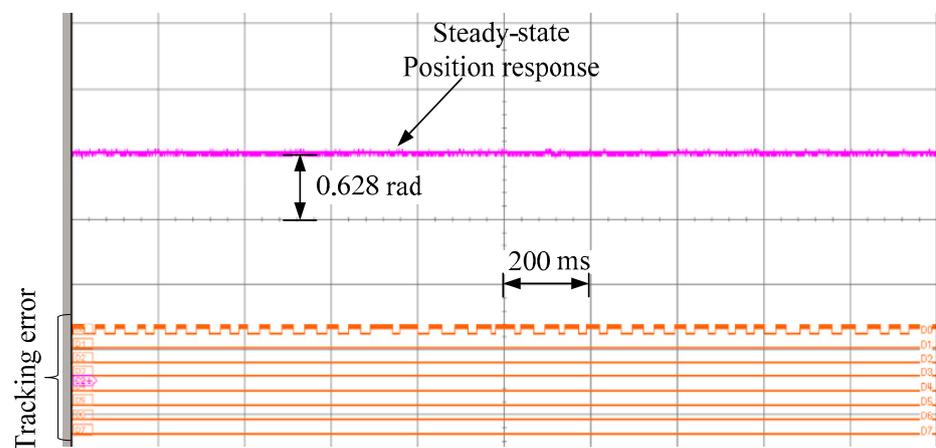


Figure 19. The experimental position steady-state error without load.

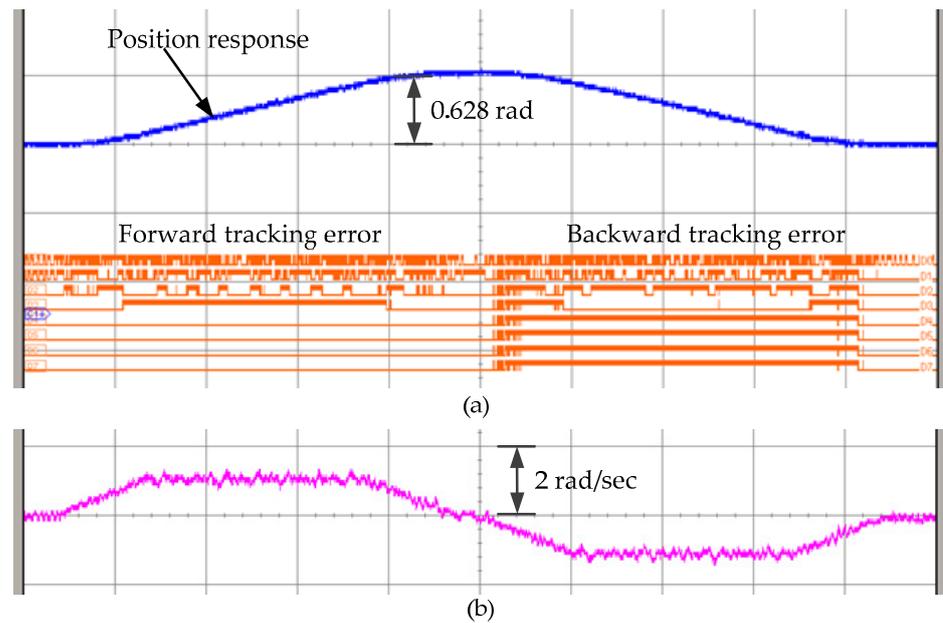


Figure 20. The responses of run forward and backward with trapezoidal velocity profile. (a) position and position tracking error; (b) speed.

Figures 21 and 22 are, respectively, the stator current and sliding surface σ . Since the system is with a trapezoidal velocity profile command input, the current response is requested to respond to the similar profile as the speed response. In Figure 22, the sliding surface is always kept at a quite low value since the system enters the sliding phase from starting.

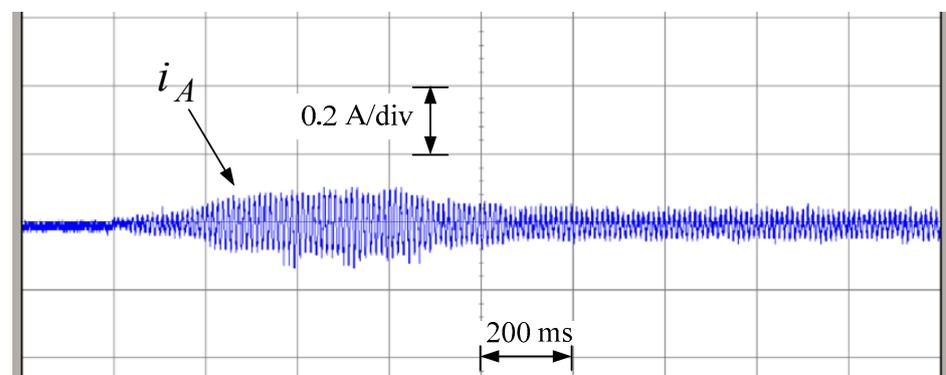


Figure 21. The current response without load.

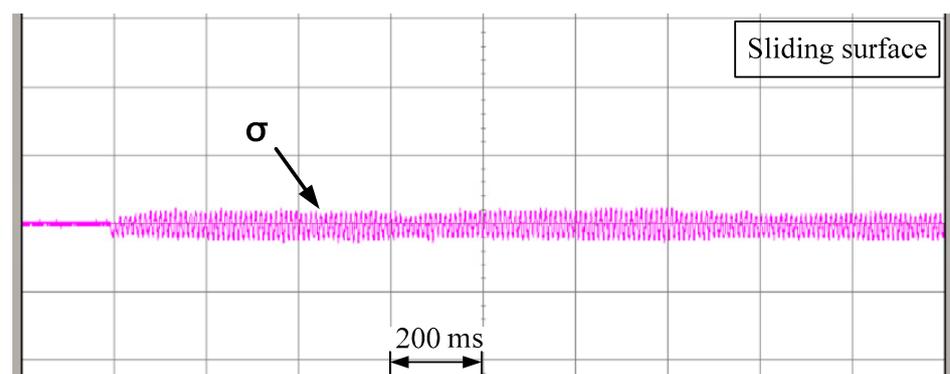


Figure 22. The sliding surface σ without the load added.

Next, we are going to display the test with load. The motor is connected to a disk load, as shown in Figure 15b, and four counterweights are added to increase the rotor inertia to about 10 times the case without them, and two-time frictions are assumed. Figures 23–25 are the results of load conditions.

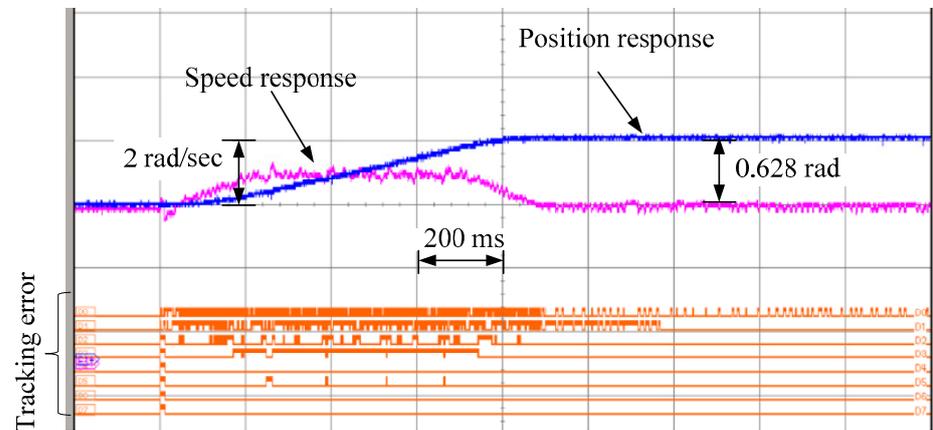


Figure 23. The position and speed responses with load.

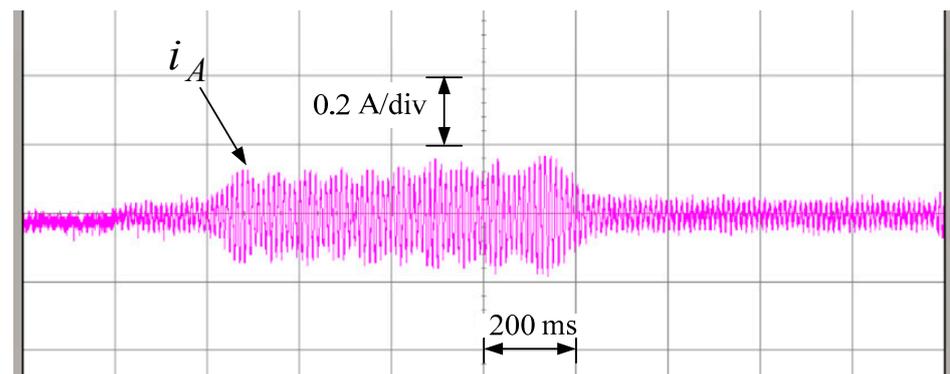


Figure 24. Current response with load.

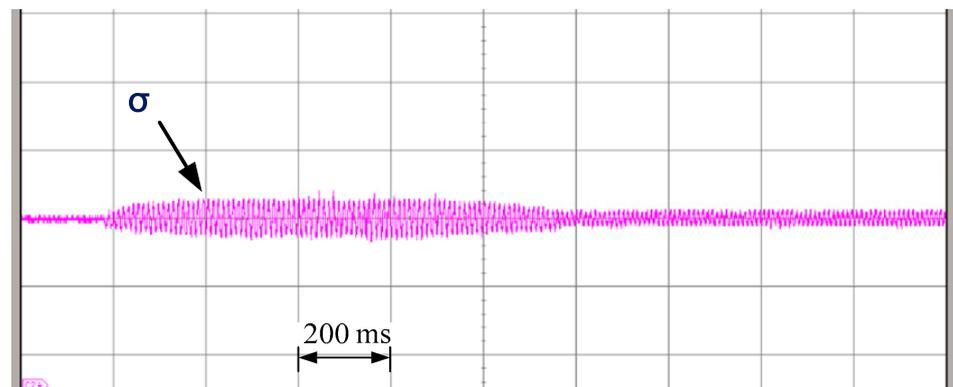


Figure 25. The sliding surface with load.

Figure 23 shows the position and speed responses under the load condition. The speed response is obedient to follow the speed profile as the no-load condition. Since the tested load is constructed by increasing the rotor inertia and friction, more generated torque is needed to overcome the variations as the evidence that the stator current was increased to about 0.35A peak to peak value as shown in Figure 24, and its magnitude is

greater than the one without load connected. Comparing the two sliding surface responses of Figures 22 and 25, they show that the ripple of the load condition is greater than the case without load. On the contrary, in the steady-state duration, the system speed is almost zero, and the increased inertia and friction lead to the reduction of system bandwidth. Thus, the magnitude of the ripple about σ for the load condition is smaller than the unload condition at the steady-state duration. It can also be seen from the steady-state digital tracking error where the load condition is not so active compared to the no-load condition.

6. Conclusions

As we can see in this paper, the stepper motor drive system with a variable structure controller implemented on an FPGA chip has been successfully realized. The system was first designed and simulated on the MATLAB/Simulink platform to evaluate the performance of the inner loop vector current control and outer loop position/speed variable structure control. It has been shown to be able to overcome the nonlinear characteristics of the fourth harmonic cogging torque and external load; the positioning error is about one bit. The hardware system implemented on FPGA for the variable structure controller and PI controller are verified by practical experiment for the stepper motor drive system with the trapezoidal velocity profile constraint, and the drive system is completed operated on the predefined profile. With the results, it shows that a DSP- or microprocessor-based control system can be replaced by a hardware controller realized by FPGA. Furthermore, as the results show in Figures 9 and 14, the system is initially in the sliding surface and the following successive time. It means that the system's characteristics are completely matched to the designed equivalent system matrix, A_c .

In the simulated speed responses of Figures 7 and 12, the speed ripples are obviously different, and no load condition is bigger than with the load condition. An estimator to dynamically determine the magnitude of q_0 and q_1 in (30) and (31) is going to be designed to reduce the chattering.

Author Contributions: Conceptualization, C.-K.L. and H.-Y.L.; methodology, C.-K.L., B.-W.L. and H.-Y.L.; software, C.-K.L., B.-W.L. and G.-Y.C.; validation, C.-K.L., B.-W.L. and G.-Y.C.; formal analysis, C.-K.L., B.-W.L.; investigation, C.-K.L., B.-W.L. and G.-Y.C.; resources, C.-K.L. and H.-Y.L.; data curation, B.-W.L. and G.-Y.C.; writing—original draft preparation, B.-W.L. and G.-Y.C.; writing—review and editing, C.-K.L. and H.-Y.L.; visualization, C.-K.L., B.-W.L. and G.-Y.C.; supervision, C.-K.L. and H.-Y.L.; project administration, C.-K.L.; All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Conflicts of Interest: The authors declare no conflict of interest.

References

- Lu, W.; Wang, Q.; Ji, K.; Dong, H.; Lin, J.; Qian, J. Research on Closed-loop Drive System of Two-phase Hybrid Step Motor Based on SVPWM. In Proceedings of the IEEE Vehicle Power and Propulsion Conference (VPPC), Hangzhou, China, 17–20 October 2016.
- Le, K.M.; Van Hoang, H.; Jeon, J.W. An Advanced Closed-Loop Control to Improve the Performance of Hybrid Stepper Motors. *IEEE Trans. Power Electron.* **2017**, *32*, 7244–7255. [[CrossRef](#)]
- Wang, C.; Cao, D. New Sensorless Speed Control of a Hybrid Stepper Motor Based on Fuzzy Sliding Mode Observer. *Energies* **2020**, *13*, 4939. [[CrossRef](#)]
- Lai, C.K.; Ciou, J.S.; Tsai, C.C. The Modelling, Simulation and FPGA-Based Implementation for Stepper Motor Wide Range Speed Closed-Loop Drive System Design. *Machines* **2018**, *6*, 56. [[CrossRef](#)]
- Ricci, S.; Meacci, V. Simple Torque Control Method for Hybrid Stepper Motors Implemented in FPGA. *Electronics* **2018**, *7*, 242. [[CrossRef](#)]
- Lee, Y.; Shin, D.; Chung, C.C. PID controller with Feedforward Low Pass Filters for Permanent Magnet Stepper Motors. In Proceedings of the 12th International Conference on Control, Automation and Systems, Jeju Island, Korea, 17–21 October 2012.

7. Shyu, K.K.; Hung, J.C. Totally invariant variable structure control systems. In Proceedings of the 23rd International Conference on Industrial Electronics, Control, and Instrumentation, New Orleans, LA, USA, 14 November 1997.
8. Defoort, M.; Nollet, F.; Floquet, T.; Perruquetti, W. Higher order sliding mode control of a stepper motor. In Proceedings of the 45th IEEE Conference on Decision & Control, San Diego, CA, USA, 13–15 December 2006.
9. Chen, M.; Kong, H.-T.; Liu, H. Conventional and Second Order Sliding Mode Control of Permanent Magnet Synchronous Motor Fed by Direct Matrix Converter: Comparative Study. *Energies* **2020**, *13*, 5093. [[CrossRef](#)]
10. Xia, X.; Zhang, B.; Li, X. High Precision Low-Speed Control for Permanent Magnet Synchronous Motor. *Sensors* **2020**, *20*, 1526. [[CrossRef](#)] [[PubMed](#)]
11. Lin, C.-H.; Hsiao, F.-Y. Proportional-Integral Sliding Mode Control with an Application in the Balance Control of a Two-Wheel Vehicle System. *Appl. Sci.* **2020**, *10*, 5087. [[CrossRef](#)]
12. Chen, T.-C.; Su, Y.-C. High Performance Algorithm Realization on FPGA for Stepper Motor Controller. In Proceedings of the SICE Annual Conference 2008, The University Electro-Communications, Chofu, Japan, 20–22 August 2008.
13. Zhang, C.; Wu, X.; Zuo, X. FPGA Soft-Core Based Step Motor Driving. In Proceedings of the 2010 International Conference on Electrical and Control Engineering, Wuhan, China, 25–27 June 2010.
14. Cerezo, J.O.; Morales, E.C.; Plaza, J.M.C. Control System in Open-Source FPGA for a Self-Balancing Robot. *Electronics* **2019**, *8*, 198. [[CrossRef](#)]
15. Hace, A. The Improved Division-Less MT-Type Velocity Estimation Algorithm for Low-Cost FPGAs. *Electronics* **2019**, *8*, 361. [[CrossRef](#)]
16. la Tona, G.; Luna, M.; di Piazza, M.C.; Pucci, M.; Accetta, A. Development of a High-Performance, FPGA-Based Virtual Anemometer for Model-Based MPPT of Wind Generators. *Electronics* **2020**, *9*, 83. [[CrossRef](#)]
17. Kocur, M.; Kozak, S.; Dvorscak, B. Design and Implementation of FPGA-Digital Based PID Controller. In Proceedings of the 15th International Carpathian Control Conference (ICCC), Velke Karlovice, Czech Republic, 28–30 May 2014.
18. Kung, Y.-S.; Tsai, M.-H. FPGA-Based Speed Control IC for PMSM Drive with Adaptive Fuzzy Control. *IEEE Trans. Power Electron.* **2007**, *22*, 2476–2486. [[CrossRef](#)]
19. Siwakoti, Y.P.; Town, G.E. Design of FPGA-Controlled Power Electronics and Drives Using MATLAB Simulink. In Proceedings of the 2013 IEEE ECCE Asia Downunder, Melbourne, VIC, Australia, 3–6 June 2013.
20. Kuo, B.C. *Automatic Control System*, 4th ed.; Prentice-Hall, Inc.: Englewood Cliffs, NJ, USA, 1982.
21. *Cyclone III Device Handbook*; Altera Corporation: San Jose, CA, USA, August 2012; Volume 1. Available online: https://www.intel.com/content/dam/www/programmable/us/en/pdfs/literature/hb/cyc3/cyc3_ciii5v1.pdf (accessed on 10 May 2021).