

## Research Article

# Implementation of High Performance Microstepping Driver Using FPGA with the Aim of Realizing Accurate Control on a Linear Motion System

Farid Alidoust Aghdam<sup>1</sup> and Siamak Saeidi Haghi<sup>2</sup>

<sup>1</sup> Mechatronics Engineering Department, Tabriz Branch, Islamic Azad University, Tabriz, Iran

<sup>2</sup> Energy Conversion Engineering Department, Mazandaran University of Science and Technology, Babol, Iran

Correspondence should be addressed to Farid Alidoust Aghdam; [info@alidoost.ir](mailto:info@alidoost.ir)

Received 1 September 2013; Accepted 27 October 2013

Academic Editors: K. Ariyur and J. Wu

Copyright © 2013 F. Alidoust Aghdam and S. Saeidi Haghi. 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 presents an FPGA-based microstepping driver which drives a linear motion system with a smooth and precise way. Proposed driver built on a Spartan3 FPGA (XC3S400 core) development board from Xilinx. Implementation of driver realized by an FPGA and using Verilog hardware description language in the Xilinx ISE environment. The driver's control behavior can be adapted just by altering Verilog scripts. In addition, a linear motion system developed (with 4 mm movement per motor revolution) and coupled it to the stepper motor. The performance of the driver is tested by measuring the distance traveled on linear motion system. The experimental results verified using hardware-in-loop Matlab and Xilinx cosimulation method. This driver accomplishes a firm and accurate control and is responsive.

## 1. Introduction

Stepper motors are increasingly used in motion control because of their low price and the ability to position well a connected electromechanical system. Mainly, stepper motors are employed in an open-loop control system, where a lot of shortcomings appear, for example, block or leakage of position by overload, torque reduction by driving in operating range, great power dissipation, and lack of robustness by dynamic loads. At the moment, to solve these problems, closed-loop systems are used; there are sensorless closed-loop controlling systems which were developed to decrease the costs of stepper motor controlling systems. Here we have only the measurement of the phase currents and no encoder which is the most expensive device in such a control system. The control algorithm can build on different kinds of microprocessors. When the closed-loop driver is employed with a DSP via C language, the coding and debugging are straightforward. For such uses it is possible to use FPGAs, where we can unite software and hardware solutions to get better performances because of the parallel computing

architecture of the FPGA [1–3]. By means of this technique, the programming and debugging are more challenging. Because of this, the performance of the FPGA-based driver is still limited. However, an FPGA-based driver with Verilog or VHDL code is easier in ASIC manufacturing. By using FPGA, it is likely to get a faster and more accurate control than it is possible with a microcontroller. Our aim is to improve a microstepping driver for stepper motors which is able to do microsteps in a stepper motor, so that effects like signs of age or the heat which influence the stepper effectiveness, have not any or very low influence on the driver. One more benefit of developing such systems in FPGA will be the power consumption. With low power FPGA devices, it is conceivable to have less power dissipation than in a microcontroller [4–8].

## 2. Microstepping Driver Structure

*2.1. Theory of Operation.* The stepper motor rotation has several direct relationships to the applied input pulses. The

TABLE 1: Duty cycle values for microstepping.

Step number	$D$
1	251
2	238
3	217
4	188
5	154
6	114
7	70
8	24
9	24
10	70
11	114
12	154
13	188
14	217
15	238
16	251

sequence of the applied pulses is directly related to the direction of motor shaft rotation. The simplest method to drive a stepper motor is controlling it by switching on/off the current through control windings. Typically, the control windings are excited through driver transistors [9, 10].

Rotating a stepper motor at its rated step size, mainly in high speed condition due to natural resonant frequencies of the motor, results in less smooth movement or in some cases leads to halt movement. Microstepping is a procedure used to smooth the motor's movement between full steps, solve resonant problems in high speed uses, and increase the step resolution of the motor. Microstepping also develops the efficiency of the system, since the current in the windings of the motor is handled in a controlled routine rather than being turned on and off sharply [9, 10]. Two main advantages of this outline have been well reported in the literature such as reduction of resonance behavior [9, 11–13] and smooth drive with very low ripple torque [11–14]. A microstepping procedure known as high torque microstepping consecutively alternates the current in the two windings of a stepper motor.

Figure 1 shows a graph of the current in the windings as opposed to angular position using this method [10]. A brief description of what is happening is that one winding is driven while the current in the other winding is dropped to zero bit by bit, reversed, and then ramped up again. This sequence then recurs for the other winding. Note that the transition between a winding being energized in one direction and then energized in the other direction has a sinusoidal shape (Figure 1). This figure offers the smoothest transition between the motor's rated step increments (i.e., 7.5 degrees). This shape is reached using a digital controller through the use of PWM. Modulating the input to the drive circuitry for a specific winding will result in a current that is related to the duty cycle of the modulated waveform. For instance, if a 5 V stepper motor is rated at 1 amp, then modulating a 5 V

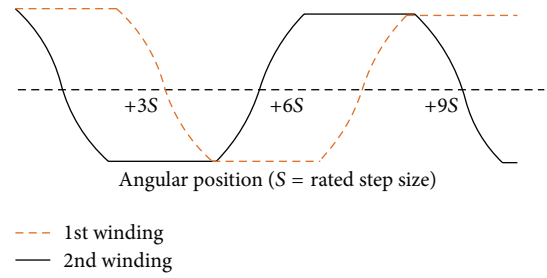


FIGURE 1: The current in the windings versus angular position.

supply across the winding at 50% will result in a current of 0.5 ampere (supposing low inductance for windings) [5–10]. The following equation shows this relationship:

$$I \approx D \times I_{\max}, \quad (1)$$

where  $I_{\max}$  is the rated current of the motor and  $D$  is the duty cycle of the PWM signal. With the aim of achieving the sinusoidal transition from a positive to negative charge in a winding, several microsteps are needed. The amount of microsteps typically ranges from 4 to 32 or more (power of 2) microsteps per rated step size. Rather than real-time computing of the duty cycle for each individual microstep, a duty cycle look-up table is realized in software. The number of table values is equal to the number of steps anticipated for a particular microstepping sequence. Equation (2) is used to attain the duty cycle values for the top half of the table. The next half of the table is basically the top half in a contrary direction [5–10]. We have

$$D(\text{step\_number}) = \frac{\cos(\text{step\_number} \times \pi)}{(\text{total\_steps} + 1)} \times (2^{\text{total\_steps}} - 1). \quad (2)$$

Using (2), duty cycle values were calculated for 16 microsteps per full step sequence using an 8-bit PWM signal in Table 1. Figure 2 shows the block diagram of a microstepping driver. The driver contains a pulse counter and sine wave LUT in the software section and power converter in hardware. To rotate the stepper motor, the currents in the phases of the motor are controlled to track sine and cosine waves. To do this, the pulse commands are counted up to get the ordered position, and the ordered position is used as an index of sine wave LUT. The output of the sine wave LUT is the reference input to the power converter and the currents in the phases of the motor are controlled to track the input reference. In other sections of the paper, the design of these blocks will be explained in detail.

There are H-bridge circuit, antialiasing filter, analog to digital converter (ADC), PI-controller, and PWM generator units in the proposed driver. The H-bridge circuit is used for transferring the current in phases of the motor in a smooth way. The ADC and antialiasing filter are used to get the sampled current value for the PI current controller. The

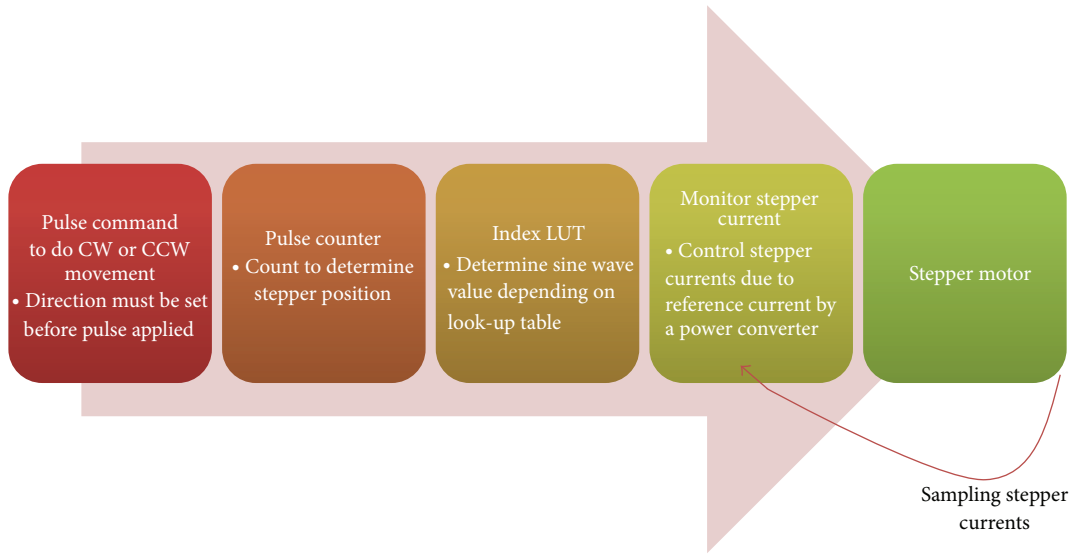


FIGURE 2: Block diagram of microstepping method.

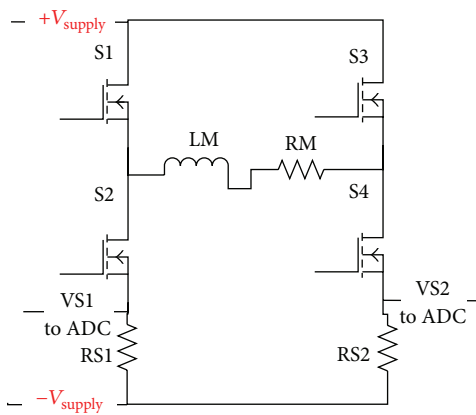


FIGURE 3: H-bridge circuit connected to stepper motor.

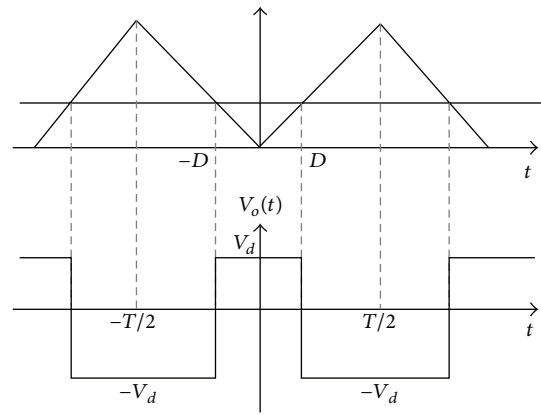


FIGURE 4: Duty cycle and voltage on a phase of stepper motor.

PI controller will adjust the duty cycle of switching pulse according to the current error.

**2.2. Bridge Circuit and Current Feedback.** The motor currents are controlled by using pulse width modulation procedure (PWM). A dual H-bridge circuit is used to drive a two-phase stepper motor. Such an H-bridge structure is shown in Figure 3. Each phase winding is coupled to each H-bridge circuit; hence, four switching devices are individually controlled to make suitable voltage for each phase winding. The leading benefit of this circuit topology is the independent generation of bipolar voltage between two phases. The H-bridge circuit (Figure 3) consists of four MOSFETs and two low resistance sensing resistors for each phase of the stepper motor [12, 14]. The four mentioned MOSFETs are controlled via a MOSFET driver IC (HIP4081A). With this H-bridge circuit, the currents flow through the phases of the motor as well as the sensing resistors RS1 and RS2. The voltages VS1 and VS2 on the sensing resistors provide current information by

the relation  $V_S = R_S * I$ . This voltage is filtered and sampled by an ADC to produce a digital value. This digital value represents the value of the feedback current [11, 14–17].

**2.3. PI Controller.** A PI controller is used to control the duty cycle of the pulses that trigger switching devices [11, 14]. The input of the PI controller is the current error between the reference current and feedback current (VS1 and VS2), and the output is the duty cycle of the switching pulses. To design such a PI controller, the first model of the PWM converter is needed. The H-bridge circuit (Figure 3) contains four MOSFETs, DC power, and the phase of the motor (LM as inductance and RM as resistance). The four MOSFETs are preserved as two pairs of switches (S1, S4 and S2, S3). The MOSFET gate drive IC (HIP4081A) will set one of these two sets of the switches to the ON state. The period and duty cycle of the switching pulse are  $T$  and  $D$ , correspondingly. Figure 4 shows the duty cycle ( $D$ ) and voltage supplied to one phase of the stepper motor. Adjusting the

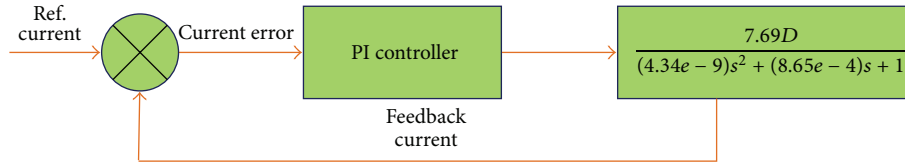


FIGURE 5: Simplified block of a power converter.

value of  $D$  allows the average DC voltage output to be varied. The phase voltage can be described as follows:

$$V_o(t) = \begin{cases} -V_d \leftarrow \frac{-T}{2} < t < -D \\ V_d \leftarrow -D < t < D \\ -V_d \leftarrow D < t < \frac{T}{2}. \end{cases} \quad (3)$$

The relation between VO and the phase current IO is derived using the model of the RL circuit [11, 14].

The phase of the motor is considered as a low pass filter [5]. The duty cycle ( $D$ ) is the output of the controller. In fact, the output of the controller is a digital value that is compared with the value of an up-down counter for the purpose of determining the switching moment of the MOSFET pairs. We have

$$I_{\text{feedback}} = \frac{7.69D}{(4.34e-9)s^2 + (8.65e-4)s + 1}. \quad (4)$$

With the transfer function as in (4), a PI controller is used for the current controlling loop Figure 5. By using *sisotool* and the Ziegler-Nichols method in Matlab software, coefficients for the PI controller are selected as the optimal values [11, 14]. The optimal coefficients of the PI controller are given by the following:

$$\begin{aligned} K_p &= 3.54e6 \\ K_I &= 9.1e-6 \\ G_{PI} &= \frac{3.54e6(1 + (9.1e-6)s)}{s}. \end{aligned} \quad (5)$$

Bode plot of the closed loop power converter is exposed in Figure 6. The PI controller will regulate the duty cycle of the switching pulses with regard to the error in stepper motor phase current. The PWM generator takes this duty cycle and creates the equivalent switching pulses to send to the HIP4082 IC, with the aim of controlling the MOSFET pair switches. Through this control outline, the output current is organized in such a way as to yield the desired current in stepper motor phases [11–14].

However, the proposed PI controller is in the analog domain; since we use FPGA which is a discrete device, then we have to use a digitized version of the PI controller. Fortunately, Matlab software has a toolbox called fixed-point converter which could help us in converting analog transfer functions to a discrete fixed-point one that could be implemented in the FPGA.

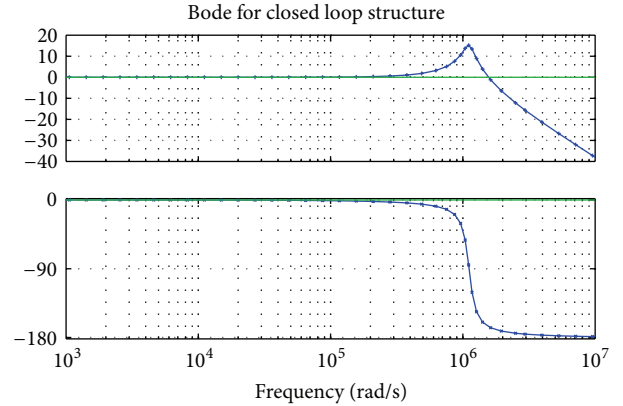


FIGURE 6: Bode plot of closed loop current controller.

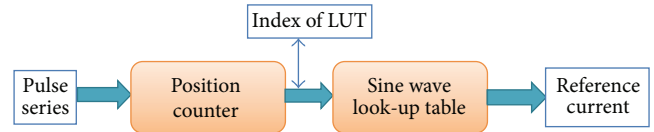


FIGURE 7: Position estimation and reference current sine wave output.

**2.4. Pulse Counter Unit and Sine Wave LUT.** The digitized sinusoidal voltage is made via the zero order hold (ZOH) module. The preceding section debates the power unit. To revolve the stepper motor, the currents in the phases of the stepper motor need to be controlled to track sine and cosine signals. The preferred position is sent to the driver by a series of pulse commands. The frequency of these command pulses depends on the desired speed. The driver uses a counter for counting the input pulses and the value of the counter is the index of the sine and cosine LUT to produce the reference current of the power converter (Figure 7) [11–14].

### 3. Hardware Structure

This section shows the implementation of the driver presented in the prior sections in the FPGA-based stepper motor driver. The experimental results prove the performance of the proposed driver.

**3.1. Hardware Configuration.** The overall driver circuit shown in Figure 8 is realized using a Xilinx Spartan3 (XC3S400 Core) FPGA development board operating at 50 [Mhz] system clock. A shot of the overall hardware setup

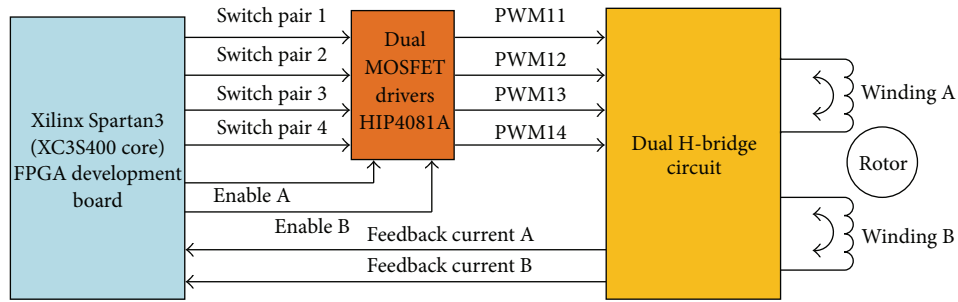


FIGURE 8: Block diagram of circuit for microstepping.

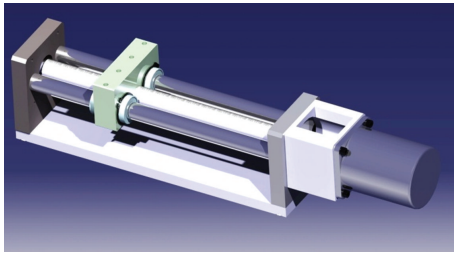


FIGURE 9: Developed linear motion system in CAD environment.

is shown in Figure 10. In this system, the switching frequency was set at 40 [Khz]. Furthermore this is the frequency of overall algorithms which are implemented. Besides, measured currents are sampled at this frequency using the AD578 12-bit ADC outside of the development board. Due to the structure of H-bridge circuit, the FPGA software is designed to generate 1[us] of the dead time for upper and lower switching device pairs to avoid short circuit in switching process. Our two-phase stepper motor is rated at 2 A and 1.8° step angle. Xilinx-ISE software is used as an FPGA development and debugging tool.

We had designed and fabricated a linear motion system that attached to the stepper motor. The structure of the linear motion system in CAD environment and its real implementation are shown in Figures 9 and 10, respectively. This motion system will travel 4 mm in forward/backward direction in each stepper motor clockwise/counterclockwise full revolution.

Figure 10 illustrates the configuration of the experimental system. The proposed FPGA-based driver includes a stepper motor and an attached load to its rotor which is a linear motion system (4 [mm] travel per motor full revolution). The control processes are implemented on an FPGA chip (XC3S400-PQ208) of Xilinx and there are other devices such as ADC (AD578) for sampling the analog input current feedback signals. The sampling frequency of the driver current controller is 40 [KHz], also the PWM switching frequency is 40 [KHz], the supply voltage is 24 [V], each phase inductance is 4.3 [mH], and the phase resistance is 5 [Ω]. Table 2 shows a summary of the utilized resources of the FPGA and Table 3 is a summary of the timing of the FPGA implementation. The input clock frequency of the FPGA board is 50 [MHz]. In this section, the performance

TABLE 2: FPGA device utilization summary.

Resources	Value
Selected device	XC3S400-4 PQ208
Number of slices	1265 out of 3584 [35%]
Number of Slice flip flops	632 out of 7168 [8%]
Number of 4 input LUTs	2038 out of 7168 [28%]
Numbers used as logic	127
Numbers used as shift registers	127
Number of IOs	264
Number of bonded IOBs	52 out of 264 [20%]
Number of MULT18X18s	6 out of 16 [38%]

TABLE 3: FPGA timing report.

Expression	Value
Minimum period	20 ns (50 MHz)
Minimum input arrival time before clock	11.374 ns
Maximum output required time after clock	7.542 ns
Maximum combinational path delay	6.373 ns

of the driver is verified. In order to verify the flowing currents in stepper motor we performed a hardware-in-loop verification using Matlab-Simulink and Xilinx-Chipscope softwares. Such a configuration is illustrated in Figure 11. First, an experiment was performed to check the performance of the current control loop. Second, the motor is controlled to perform a smooth motion. Figure 12 shows the reference current and feedback current in one phase of the motor. The reference current is a sine wave and the output current follows a reference sine wave. A small ripple (about 40 [mA]) is seen in the output current due to current switching components (Figure 13). The maximum current is set to 2 [A]. The phase shift between the two currents is caused by delays in the driver control loop. This result shows that the current output can follow the desired current. This is because the position of the motor depends on the value of the current; when the output current is smooth, the precise position control can be achieved. In this FPGA-based experimental system, the motor driver can perform position control with maximum 2 pulse errors in 16 microsteps mode, which results in 2.5 [μm] error in developed linear motion system.



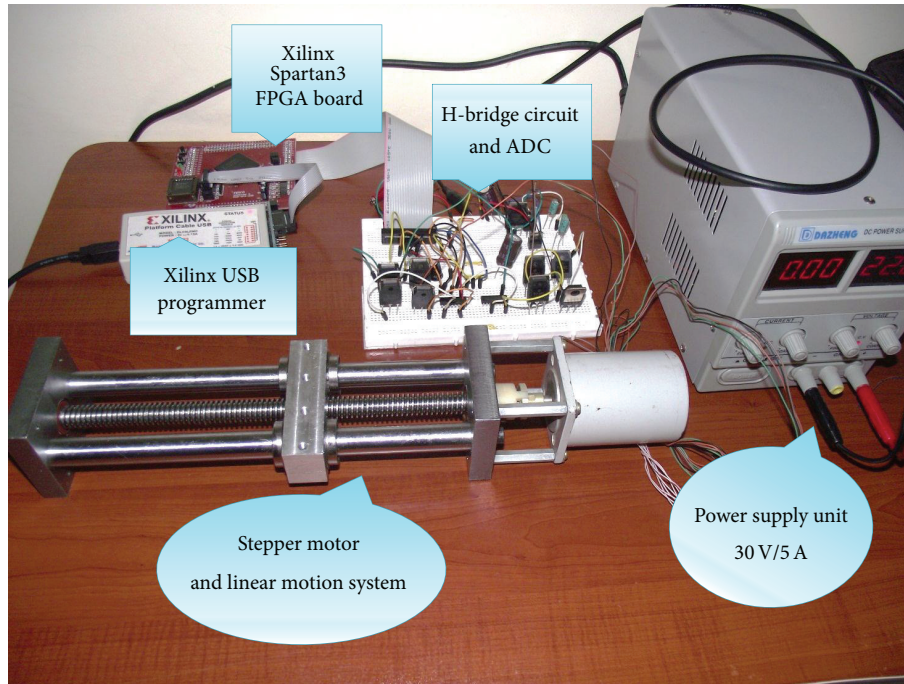


FIGURE 10: Overall real hardware setup.

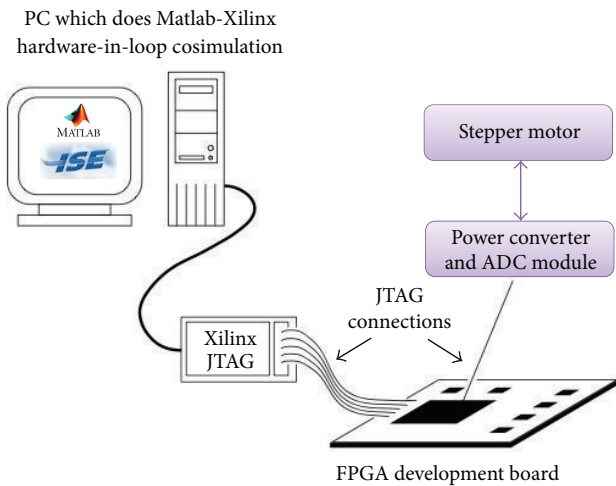


FIGURE 11: Hardware-in-loop cosimulation.

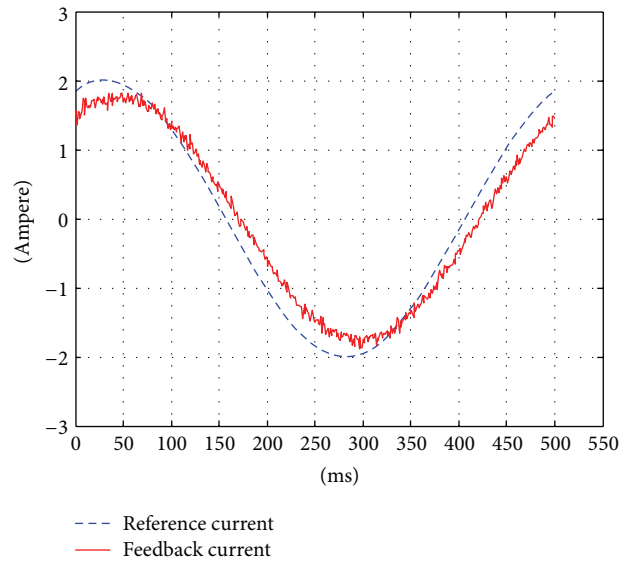


FIGURE 12: Reference current and feedback current in 500 [ms].

### 4. Results

With respect to the achieved results, the efficiency of the driver is confirmed. First, an experiment was performed to check the efficiency of the driver control loop. Second, the motor is controlled to follow a smooth motion. Figure 12 shows the reference current and feedback current in a phase of the stepper motor. The reference current is a sine wave and the output current follows this sine wave. The maximum current is 2 [A]. The phase shift between the two currents is caused by the delays existing in the driver control loop and the mechanical movements. This result shows that the current output can follow the desired current. This is because the

position of the motor depends on the value of the current; when the output current is smooth, the precise position control can be achieved. Figure 13 shows the currents in the 2 phases of the motor when the motor is controlled at a constant speed. Thus, the currents in the 2 phases follow a sine wave and shift in phase by an electric angle of 90° (this is a cosine wave). Tables 2 and 3 represent the results of the implementation of the microstepping driver on the FPGA device. In this FPGA-based experimental system, the driver can perform position control with 2.5 [ $\mu$ m] travel error or 2

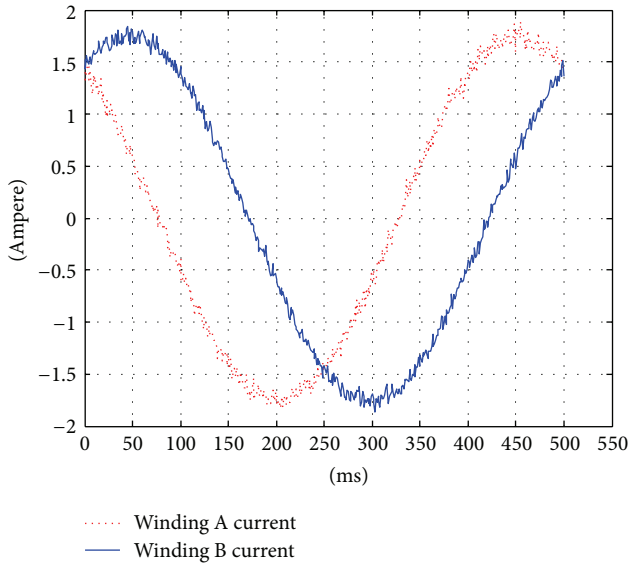


FIGURE 13: Sampled currents in 2 phases of stepper motor in 500 [ms].

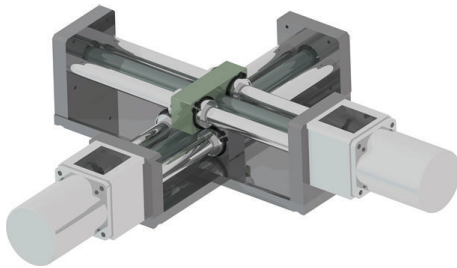


FIGURE 14: Combination of multiple linear systems to make multiple axis structures.

pulse error when stepper motor drive linear motion system in a 20 [cm] distant. With this precision, it is possible to make small scale, and low cost, low complexity multiaxis structures. Such a construction is presented in Figure 14.

## 5. Conclusion

Stepper motors are used in an extensive range of position controlling devices such as robots, printers, plotters, and CNCs. Nowadays, most of the stepper motor controlling systems do not work as accurately as they have potential to. Microstepping is a drive procedure for the stepper motors that allows the smooth movement of the rotor in a fraction of the motor's full step angle. In high precision applications, the microstepping driver is essential for accurate motor spin. In this paper a digital implementation of a microstepping drive system with current control using a Xilinx Spartan3 (XC3S400) FPGA development board was presented. The dual H-bridge converter is planned to drive the two-phase stepper motor. The model of the current controller and gains of the PI controller for optimal control are calculated using Matlab software. In our design, the FPGA is used to construct a high performance microstepping driver without using any

microcontroller, DSP, or driver chip; additionally because all the algorithms are written in Verilog hardware description language, it is compatible with migrating proposed scheme to an ASIC and bulk fabrication methods. The fractional step can be configured and the rotor movement can be controlled up to 1/16-step resolution giving an overall resolution of developed linear motion system. Experimental results tested using hardware-in-loop Matlab and Xilinx cosimulation method. Finally, the experimental results validate the performance of the driver.

## References

- [1] Z. Xiaodong, H. Junjun, and S. Chunlei, "An approach of micro-stepping control for the step motors based on FPGA," in *Proceedings of the IEEE International Conference on Industrial Technology (ICIT '05)*, pp. 125–130, December 2005.
- [2] T. Takahashi and J. Goetz, "Implementation of complete AC servo control in a low cost FPGA and subsequent ASSP conversion," in *Proceedings of the 19th Annual IEEE Applied Power Electronics Conference and Exposition (APEC '04)*, pp. 565–570, February 2004.
- [3] S. Xiaoyin and S. Dong, "Development of an FPGA-based motion control ASIC for robotic manipulators," in *Proceedings of the 6th World Congress on Intelligent Control and Automation (WCICA '06)*, pp. 8221–8225, June 2006.
- [4] H. K. Bae and R. Krishnan, "A study of current controllers and development of a novel current controller for high performance SRM drives," in *Proceedings of the 31st IEEE Industry Applications Society Annual Meeting (IAS '96)*, pp. 68–75, October 1996.
- [5] A. I. Maswood, "PWM voltage source inverter with PI controller for instantaneous motor current control," in *Proceedings of the 1995 International Conference on Power Electronics and Drive Systems*, pp. 834–837, February 1995.
- [6] P. Enjeti, J. F. Lindsay, P. D. Ziogas, and M. H. Rashid, "New current control scheme for PWM inverters," *IEEE Proceedings B*, vol. 135, no. 4, pp. 172–179, 1988.
- [7] S. Halasz, G. Csonka, A. A. M. Hassan, and B. T. Huu, "Analysis of the unipolar PWM techniques," in *Proceedings of the 1996 8th Mediterranean Electrotechnical Conference (MELECON '06)*, pp. 353–356, May 1996.
- [8] S. Halasz, "Analysis of pulsewidth modulation techniques for induction motor drives," in *Proceedings of the IEEE International Symposium on Industrial Electronics (ISIE '93)*, pp. 200–204, Budapest, Hungary, 1993.
- [9] T. Kenjo and A. Sugawara, *Stepping Motors and Their Microprocessor Controls*, Clarendon Press, Oxford, UK, 1984.
- [10] R. Condit, *Stepper Motor Control Using the PIC16F684*, Technology Inc., 2004.
- [11] Q. L. Ngoc and W. J. Jae, "An open-loop stepper motor driver based on FPGA," in *Proceedings of the International Conference on Control, Automation and Systems (ICCAS '07)*, pp. 1322–1326, October 2007.
- [12] S.-M. Yang and E.-L. Kuo, "Damping a hybrid stepping motor with estimated position and velocity," *IEEE Transactions on Power Electronics*, vol. 18, no. 3, pp. 880–887, 2003.
- [13] Z. Qiu, S. Shi, X. Li, L. Zhang, and W. Wu, "Implementation of motion control technique for stepper motor translation stages in online detection system," in *Proceedings of the 2011 International Conference on Mechatronic Science, Electric Engineering and Computer (MEC '11)*, pp. 71–75, August 2011.

- [14] N. Dahm, M. Huebner, and J. Becker, "Approach of an FPGA based adaptive stepper motor control system," in *Proceedings of the 6th International Workshop on Reconfigurable Communication-Centric Systems-on-Chip (ReCoSoC '11)*, pp. 1–6, June 2011.
- [15] Z. Ali and R. V. Kshirsagar, "Development of a CPLD based novel open loop stepper motor controller for high performance using VHDL," in *Proceedings of the 14th International Conference on Intelligent Engineering Systems (INES '10)*, pp. 307–312, May 2010.
- [16] D. Rossi and A. Cuomo, "Analog multiplex for sensing the magnitude and sense of the current through a h-bridge stage utilizing a single sensing resistance," Google Patents, 1989.
- [17] J. D. Wale and C. Pollock, "A low-cost sensorless technique for load torque estimation in a hybrid stepping motor," *IEEE Transactions on Industrial Electronics*, vol. 46, no. 4, pp. 833–841, 1999.





**Hindawi**

Submit your manuscripts at  
<http://www.hindawi.com>

