IEEE TRANSACTIONS ON INDUSTRIAL ELECTRONICS, VOL. 55, NO. 4, PP. 1548-1557, APRIL 2008

Control System with High-Speed and Real-Time Communication Links

Tianjian Li and Yasutaka Fujimoto, *Member, IEEE*

***Abstract*—Recent technological advances have enabled distributed control systems to be implemented via networks. This allows feedback control loops to be closed over communication channels. This paper develops a control system with high-speed and real-time communication links. Two-degrees-of-freedom control is utilized in this servo control system and delta-sigma modulation is employed to compress data and to transmit the signal over the transmission channels between the controller and the controlled plant. Simulation results show that the controller can compensate for noise in signal transmission channels. In addition, the developed system is implemented in field-programmable gate arrays (FPGA). We developed a real-time closed-loop control system with a communication channel whose control-sampling period is 600 ns, and can reduce the sampling period of the controller module to hundreds of nanoseconds.**

***Index Terms*—Networked control, Real-time control, Delta-sigma modulation, FPGA.**

I. INTRODUCTION

Continuous improvement in automation and the growth of production scales in manufacturing have caused communication technology to be used increasingly in control systems. Real-time control networks are an emerging technology that is attracting increasing attention. From a control systems perspective, in a distributed system, the control loop formed by the plant's sensors, controllers, and actuators is closed via communication channels in real time. Control-networking technologies provide several benefits in linking data points. Networks enable remote data transfers and data exchanges among users, reduce the complexity of wiring connections and the cost of media, and provide ease in maintenance [1], [2]. Applications using such networks include robotic systems [3], bilateral teleoperation system [4], networks in automobiles [5], aircraft, and manufacturing plants.

For instance, in a robot system, there are dozens or even hundreds of controlled nodes. Obviously, there is large amount of control information transmitted between the controllers and actuators to control the robot motion accurately in real time. Therefore, we are interested in how the information is accurately and instantly transmitted, compressed, and exchanged to reduce the cost of cables and interfaces and to make system monitoring and maintenance easier. Compensating for the noise over the communication channels accurately must also be considered. Previously, numerical controllers (NC) and drive technology achieved a high standard of performance with controlled analog drives. Although position control, along with the drive, forms a functional closed control loop, position control is conventionally carried out in the NC, because analog drives do not have the necessary digital processing capability. This standard analog drive has ensured problem-free interaction between controllers and drives from different manufacturers. However, it slows system response and limits system performance. Today, the entire signal processing of a digital drive can be done with a microcontroller. The microcontroller enables not only conventional torque and speed control, but also fine-position control with extremely short cycle times. For instance, the recent SERCOS interface (SERCOS-Ⅲ can provide a 31.25 μs cycle time [6]) is clearly ahead of other standardized interfaces like PROFIBUS or CAN. The system requires a high-speed control sampling period and a suitable digital interface.

Current research on networked control systems is proceeding rapidly and in many directions. Several researchers aim at reducing the amount of information transmitted over the communication channels with data-compression methods [7]. Several recent papers have focused on the quantization problem caused by the limited channel rate [8]–[13]. Other researchers have focused on developing a new control theory under communications constraints [12], [13], and yet other researchers have focused their efforts on implementing low data-rate systems for control purposes [10]. The problem of linear-system stabilization under sampled encoded measurements has been studied [8], [9]. This is based on a dynamically adjustable "zoom-in/zoom-out" quantizer. More recently, time-invariant memoryless logarithmic-quantizers have also been proposed [10]. Their networked control systems employ micro- or millisecond control sampling periods. The fastest existing system features a 31.25μs cycle time for eight nodes with 8-byte cyclic data, under a 100Mb/s communication data rate [6]. It is difficult to shorten the sampling period because of the limited data rate. However, for robotic systems or high-speed and high-precision multi-axis drive systems, more drives and higher sampling periods are required to assure reliability and to provide the desired performance. Therefore, we have studied a low-cost and high-bit resolution quantizer to satisfy the

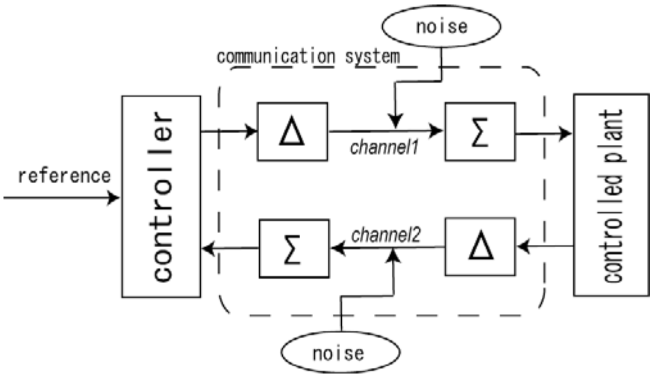


Fig. 1. Configuration of the control system with communication links.

needs of nanosecond sampling systems, with high information-exchange rates under the limited data rate. This can help us obtain better control performance in multi-node systems. This paper proposes modified delta-sigma modulation to coding source, within the context of networked control systems. Although delta-sigma modulation is well known for reducing the data rate for voice communication, it is not widely used in the context of networked control systems. The fundamental difference between the proposed delta-sigma modulation and the classic version is a three-level (1, 0, −1) dynamic quantizer, rather than the conventional two levels (1, −1), which provides high compression rates, and reduced code length. Based on delta-sigma modulation we propose a novel method to compensate for bit error over noisy communication channels.

Recent advances in microprocessor technology make it possible to realize an almost ideal real-time feedback-control system with a small compensation time for calculation of the control law. The emergence of FPGA has given rise to high-performance controllers. An FPGA-based control system can dramatically reduce execution time through parallel architectures, which increase the availability of I/O channels, and decrease the cost [14], [15], [16]. These benefits have caused FPGAs to be applied in various fields, such as telecommunication [17], signal processing [18], [19], [20], embedded-control systems [21], robotics [22], [23], [24], electrical machine control system [25]–[30] and motion control [4], [31], [32]. In this paper, we focus on networked control systems implemented with FPGAs—the entire control and communication system are implemented in FPGAs.

This paper shows the development of a high-speed and real-time control/communication system, with communication channels connecting the controller and the controlled plant. The communication system uses delta-sigma modulation to compress the transmitted data. The system gains communication efficiency through this modulation method, and can successfully compensate for noise over the communication channels by setting the compensation channel and the designed servo controller. The system we implement in FPGAs can reduce the sampling period of controller scheme to the order of nanoseconds, and meet the real-time control system requirements.

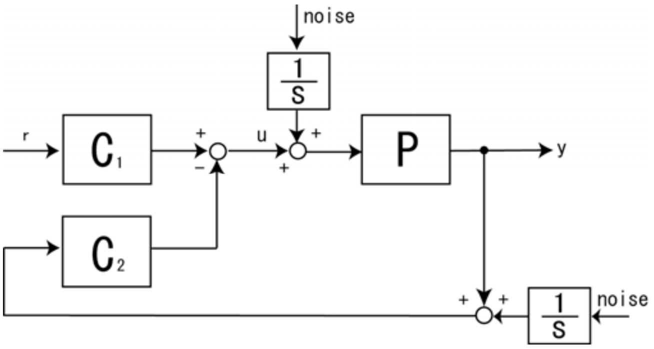


Fig. 2. Configuration of the two-degrees-of-freedom control system.

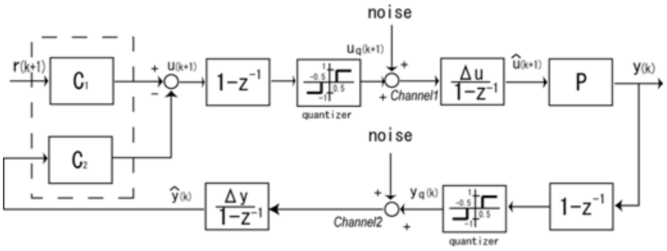


Fig. 3. Configurations of control system with delta-sigma modulation in Z- transform.

II. System Configuration

The structure of the control system with a communication link is shown in Fig. 1. The controlled plant is represented by a servomotor. Compensation for noise over the channels is designed into this system.

A. Controller Model

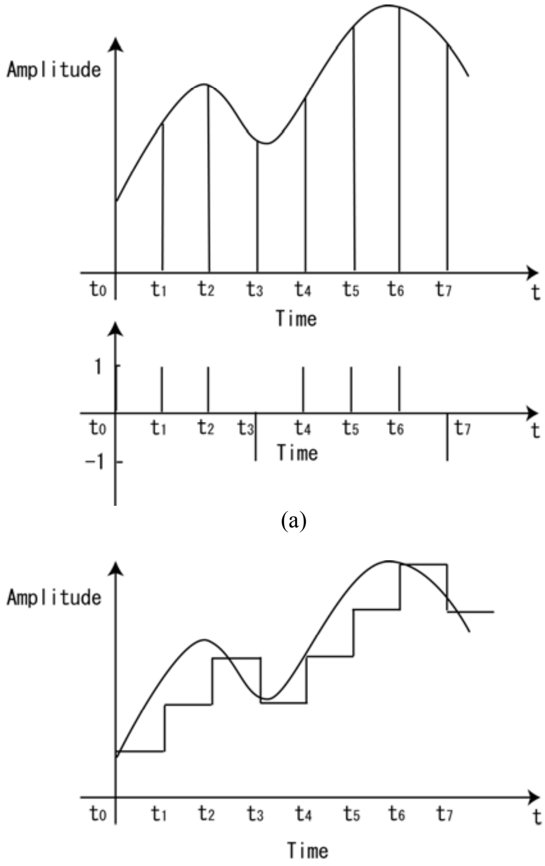
In the control system, we utilize a type 1 servo controller that can compensate for step disturbances on the input. We employ two-degrees-of–freedom controller [33], [34] to realize the developed control system. The control system configuration is shown in Fig. 2. The transfer function can be simply given as

$$u = \begin{bmatrix} C_1 & C_2 \end{bmatrix} \begin{bmatrix} r \\ y \end{bmatrix}, \tag{1}$$

where u is the controller output, the controller functions are expressed by C_1, C_2, where r is the command input, and y is the plant output.

B. Communication System Design

To perform remote closed-loop control, we utilize delta-sigma modulation to compress data in the communication system. Delta-sigma modulation has been used in numerous signal-processing applications, including A/D and D/A converters, FIR and IIR filters, AM/FM modulators, correlators, multipliers, frequency synthesizers, and synchronizers [18], [35], [36]. Using delta-sigma modulation,



(a)

(b)

Fig. 4. Function of modulation method. (a) Function of delta modulation. (b) Function of sigma modulation.

various signal-processing application circuits can be designed to achieve high-bit resolution with low hardware cost. It can convert the large bit-width data format to a 1-bit format, reducing the traffic in the communication channels. The delta-sigma modulated data stream is low-pass filtered to reduce the quantization noise, thus, it can meets the control and communication system's requirements for high-speed sampling and real-time operation. Fig. 3 shows the configuration of a communication system using delta-sigma modulation in Z-transform domain. The proposed delta-sigma modulation is a three level (1, 0, −1) quantizer.

1) *Delta modulation*

Delta modulation is based on quantizing the signal change from sample to sample, rather than the absolute value of the signal at each sample. The function is shown in Fig. 4(a) where the equation is:

$$u_q(k) = \begin{cases} 1 & if \quad u(k) - u(k-1) \geq \Delta u \\ 0 & if \quad |u(k) - u(k-1)| < \Delta u \\ -1 & if \quad u(k) - u(k-1) \leq -\Delta u \end{cases}, \qquad (2)$$

where Δu is a quantum unit of the discrete system model. The rounding off module is realized by a quantizer, which is shown in Fig. 3.

TABLE I
MOTOR SPECIFICATIONS AND THE CONTROLLER PARAMETERS

	Rated Voltage	100 [V]
	Rated Current	0.66 [A]
R_a	Rotor Resistance	$2.8\,[\Omega]$
L_a	Rotor Inductance	$0.0011\,[H]$
J_l	Motor Inertia	$4.96e^{-6}\ [N \cdot m \cdot s^2]$
D_l	Friction Coefficient	$3.543e^{-3}\ [N \cdot m \cdot s]$
K_T	Torque Const.	$0.157\ [N \cdot m / A]$
K_e	CEMF Const.	$0.157\ [V / rad / s]$
τ	Time Const.	50 [ms]
ω_c	Damped Frequency	80 [rad/s]
ξ	Damping Coefficient	0.9
Δu_{in}	Quantization (input)	10/256 [V]
Δu_{out}	Quantization (output)	0.5π/4096 [rad]
	Gear radio	0.02

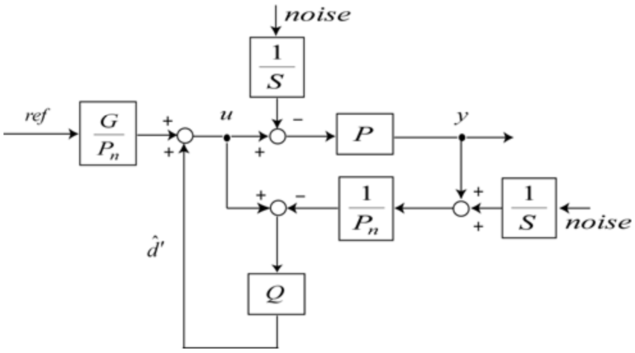


Fig. 5. Disturbance observer type equivalent block diagram

2) *Sigma modulation*

Sigma modulation is the integrator that adds and subtracts the value from the output of delta modulation. The function is shown in Fig. 4(b), where the equation is

$$\hat{u}(k) = \hat{u}(k-1) + \Delta u \cdot u_q(k)\,. \qquad (3)$$

III. APPLICATION TO SERVO MOTOR CONTROL

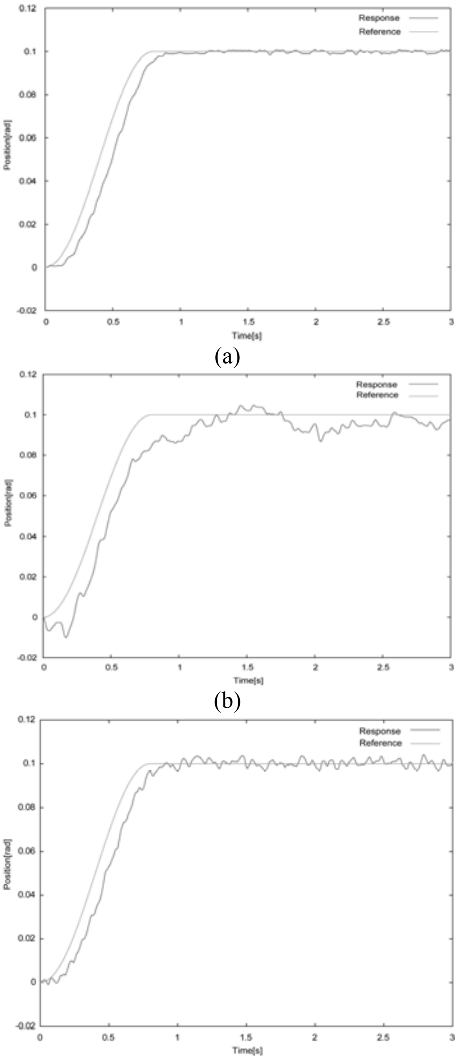
In this paper we treat the controlled plant as a servo motor to verify the developed system by numerical simulation. The nominal model of the controlled plant is expressed as

$$P_n = \frac{K_n}{(J_l s + B_n)s}\,, \qquad (4)$$

where B_n and K_n are

$$B_n = \frac{K_e K_T + R_a D_l}{R_a}\,, \qquad (5)$$

$$K_n = \frac{K_T}{R_a}\,, \qquad (6)$$



(a)

(b)

(c)

Fig. 6. Simulation results. (a) System response with noise over channel1. (b) System response with noise over two transmission channels. (c) System response using noise compensation over three channels.

where R_a is the rotor resistance, K_e is the CEMF constant, K_T is the torque constant, D_l is the friction coefficient, and J_l is the motor inertia. We designed the command-input response of the closed-loop system to be

$$G = \frac{1}{\tau^2 s^2 + 2\xi\tau s + 1}, \tag{7}$$

and the complementary sensitivity function to be

$$Q = \frac{(2\omega_c s + 1)\omega_c^2}{\left(s^2 + \omega_c s + \omega_c^2\right)(s + \omega_c)}, \tag{8}$$

where τ is the response-time constant, and ω_c is the damped frequency. This provides the following controllers [5]:

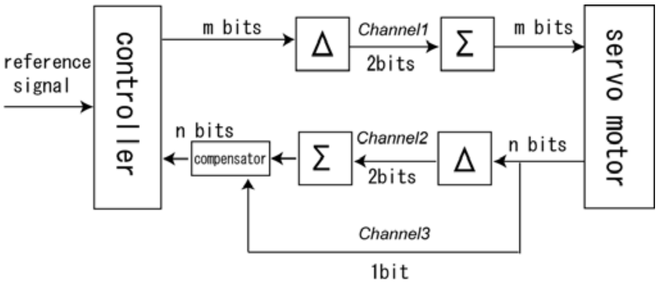


Fig. 7. Noise-compensation model of channel2.

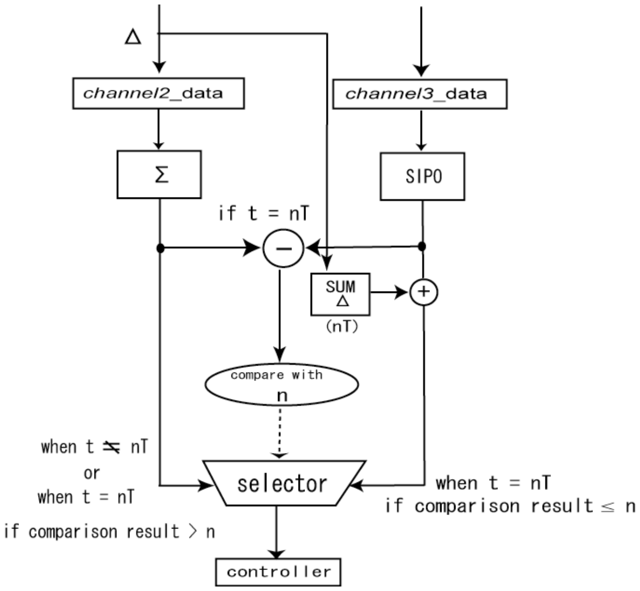


Fig. 8. Noise-compensation method in feedback channel.

$$C_1 = \frac{G}{P_n(1-Q)} = \frac{(B_n + J_l s)\left(s^3 + 2s^2\omega_c + 2s\omega_c^2 + \omega^3\right)}{K_n s(s + 2\omega_c)\left(1 + s^2\tau^2 + 2s\tau\xi\right)}, \tag{9}$$

$$C_2 = \frac{Q}{P_n(1-Q)} = -\frac{(B_n + J_l s)(2s + \omega_c)\omega_c^2}{K_n s(s + 2\omega_c)}. \tag{10}$$

The motor's specifications and the controller's parameters are shown in Table 1. The input-channel quantization value Δu_{in}, and the output channel quantization value Δu_{out}, are set to 10.0/256 V and 0.5π/4096 rad, respectively. The simulation is written in C. To verify the resistance to noise, we set the noise to be about 3% bit error over the communication channels. For example, the switching period of the pulse-width modulator (PWM) inverter is 20 μs (50 khz carrier frequency) and that of the position-control system is 0.6 μs. With this condition, the bit error will occur at about 3% level, because of PWM switching noise. This is a realistic noise level for real conditions. The two-degrees-of-freedom control system (shown in Fig. 2) can be transformed to Fig. 5 [33]. Hence, we can observe the equivalent-input disturbance caused by the noise. Thus, it is possible to estimate and compensate for the noise by the disturbance observer. The type 1 servo controller C_2 has this function.

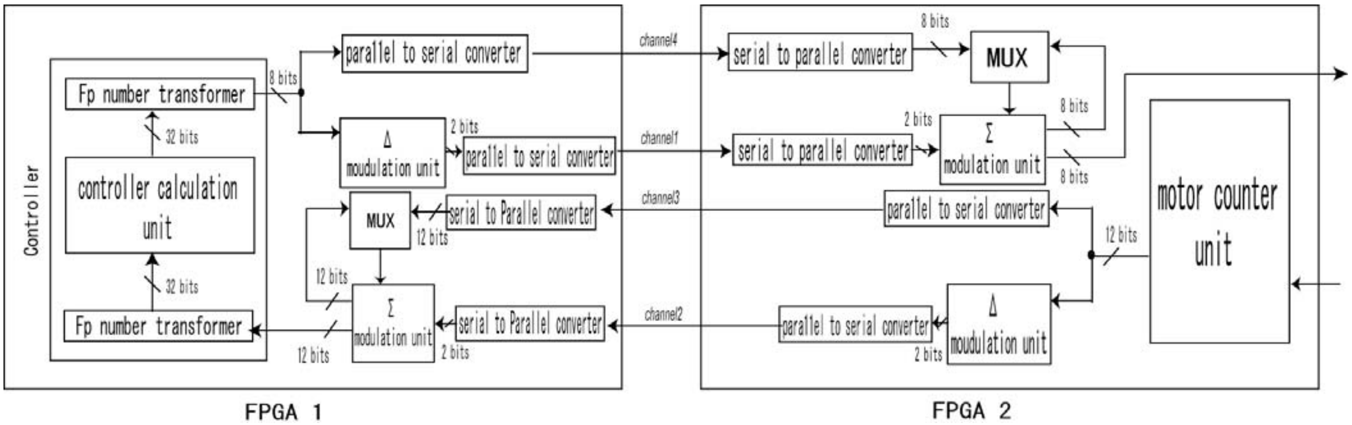


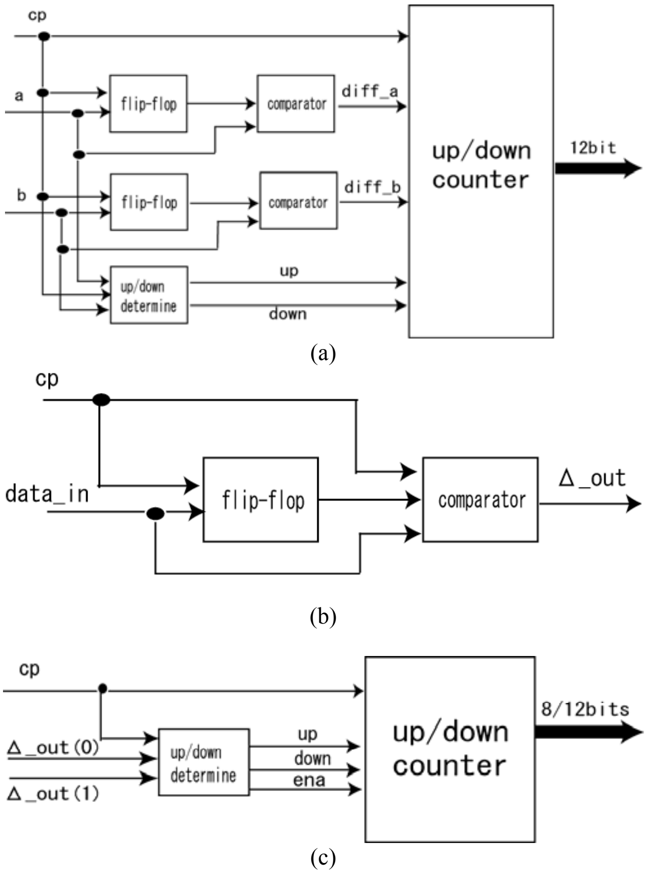
Fig. 9. The basic configuration in FPGAs.

The simulation results for the noise over channel1 are shown in Fig. 6(a). The simulation results for noise over channel1 and channel2 are shown in Fig. 6(b). Fig. 6(a) shows that the controller can compensate for noise over channel1, because controller C_2 has an integrator. However, as shown in Fig. 6(b), the system does not compensate for noise over channel2. Theoretically, the noise over channel2 is not observable. Thus, we have added channel3 to compensate for noise over channel2. The system configuration is shown in Fig. 7. Because multi-bit format digital signals can be shaped into a 1-bit format with delta-sigma modulation, the wiring costs are much lower than in some parallel communication schemes. Here**,** an *n*-bit parallel output signal from a servomotor side is transformed to a 1-bit serial signal every *n* sampling periods, and the controller side receives a 1-bit signal from channel3 to be transformed into an *n*-bit parallel signal in the same *n* sampling periods. Thus, the controller side restores the signal transmitted from servomotor. However, if channel3 is affected by noise, the restored signal will be compared with the output of the sigma modulation on the controller side in the same sampling period, and we will choose the best signal as the feedback signal from servomotor. Fig. 8 and equation (11) show the compensation process for noise over channel3. Here, the maximum variation every nth sampling period is assigned as the threshold. So, we choose a threshold, *n,* that is the same as the bit-width in channel3. Figure 6(c) shows the results when noise affects all channels—clearly, that the system can compensate for noise over all channels.

$$u(t)=\begin{cases} u_{channel2}(t) & if \quad t=nT \quad and \quad \left|u_{channel3}(t)-u_{channel2}(t)\right|>n \\ u_{channel3}(t) & if \quad t=nT \quad and \quad \left|u_{channel3}(t)-u_{channel2}(t)\right|\le n \\ u_{channel2}(t) & if \quad t\ne nT \end{cases} \cdot \tag{11}$$

IV. FPGA IMPLEMENTATION

FPGAs are one type of the programmable IC. We can implement the digital circuit in FPGAs using a hardware description language (HDL) [37]. In this paper the communication and two-degrees-of-freedom control system are implemented in FPGAs [38], [39]. The control system is described in VHSIC hardware description language (VHDL). The basic structure of the designed circuits in FPGAs is shown in Fig. 9. On left side, FPGA1 is the main controller module, one part of the delta-sigma modulation module, and the communication-interface circuits. On the right side, FPGA2 is the motor-counter module, the other part of the delta-sigma modulation, and communication-interface



(a)

(b)

(c)

Fig. 10. Communication-system circuits in FPGA. (a) Motor-counter unit. (b) Delta modulation unit. (c) Sigma modulation unit

circuits.

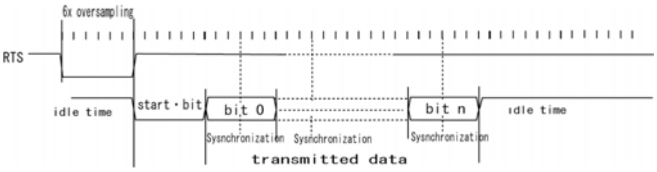


Fig. 11. Configuration of transmitted data

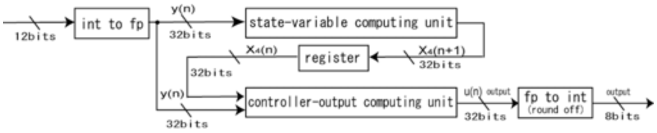


Fig. 12. Block diagram of the controller in FPGA

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	······	0
s	e								f									

Fig. 13. Thirty-two-bit floating-point number format.

To implement this system in the experiment we added channel4 to transform the output of the controller in the same way as channel3.

A. Communication System

The communication-system circuits in the FPGA are divided into three main parts:

1. Motor-counter unit.
2. Delta-sigma modulation unit.
3. Communication-interface unit.

1) Motor-counter unit

We designed the 4x multiplication counter circuit to receive and count the motor-position signals. The motor-counter unit consists of an up/down counter, a comparator, and a counter up/down determine unit as shown in Fig. 10(a), where *cp* is the clock pulse. In the counter circuit, the comparator tests the changes of the pulse of *a* phase and *b* phase form the servomotor encoder in each clock pulse. The output of comparator is 0 or 1, to trigger the up/down counter. In addition, the motor-rotation direction can be determined by the relation of *a* and *b* phase, by the up/down determine unit [38].

2) Delta-sigma modulation unit

Fig. 10(b) shows the configuration of delta modulator in FPGA, where *cp* is the clock pulse. Here, the comparator can test changes in value of the input signal, which is latched by a flip-flop at each sample. The output of comparator has three states, expressed as 2 bits—11, 00, and 01. The configuration of sigma modulation in the FPGA is shown in Fig. 10(c), where *cp* is the clock pulse. The counter will be up or down, according to the output of the delta modulator, and the value is expressed in a 12-bit signal as the output of sigma modulation.

3) Communication-interface unit

We designed the unidirectional serial-communication interface circuit with FPGAs to reduce the wiring cost. The parallel-to-serial converter is the main part in the unit. There are two links between the receiver and the driver—one carries the transmitted data stream and the other carries the RTS (Request To Send) signal. When the data is ready to be transmitted from the driver, the RTS signal is set to activation. When the RTS signal is detected as activated, the receiver is ready to receive data immediately. To read the transmitted data accurately, a 6x over-sampling method is utilized to provide more detailed sampling periods than that of a single bit. The configuration of the transmitted data is shown in Fig. 11. With this communication method, we can synchronize both FPGAs to the same clock level, providing a high communication data rate over short distances.

B. Control System

FPGA is a generic computational platform, with parallel processing and run-time reconfiguration capabilities. Linear-control system algorithms are generally multiply-accumulate-intensive functions. They contain feedback loops and parallel structures. These parallel structures or repetitive loops can be divided into multiple data paths, and processed as parallel routines in FPGA to increase the system processing speed. FPGAs are candidates for high sample-rate linear-controller implementations. Hence, we implemented a two-degrees-of-freedom controller in FPGA. The controller is designed for angle-position control of a servomotor [39]. The controller consists of three main parts, which are shown in Fig. 12.

1. Integer and floating-point number transform unit.
2. State-variable computing unit.
3. Output computing unit.

The data format for storing floating-point numbers in the controller is the IEEE 754 [40] standard. Single precision floating-point uses the 32-bit IEEE 754 format shown in Fig. 13.

In the controller module, the state-variable computing unit and the output-computing unit consist of the arithmetic logic unit (ALU), which includes addition/subtraction and floating-point multiplication. [41]

Floating-point multiplication is much like integer multiplication. Because floating-point numbers are stored in sign-magnitude form, the multiplier needs only to deal with unsigned integer numbers and normalization. Like the architecture of the floating-point adder, the floating-point multiplier unit is a three-stage pipeline that produces a result on every clock cycle.

Because we used an 8-bit DA converter and 12-bit counter, it is necessary to design a way to transform from single-precision floating-point numbers to 8-bit integers and from 12-bit integers to single-precision floating-point numbers [39]. As the first conversion step floating point to 8-bit integer, the value of right shift is computed. The second step shifts the mantissa to right. The mantissa is adjusted, and rounded off to output. For converting from 12-bit integers, the mantissa of the 12-bit input data is separated, and an exponent is built, the exponent adjusted and the mantissa is output.

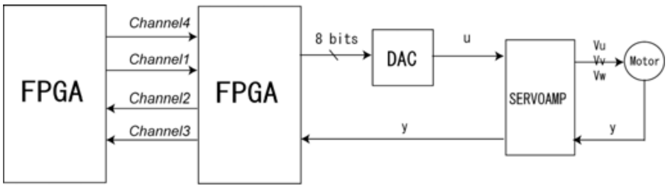
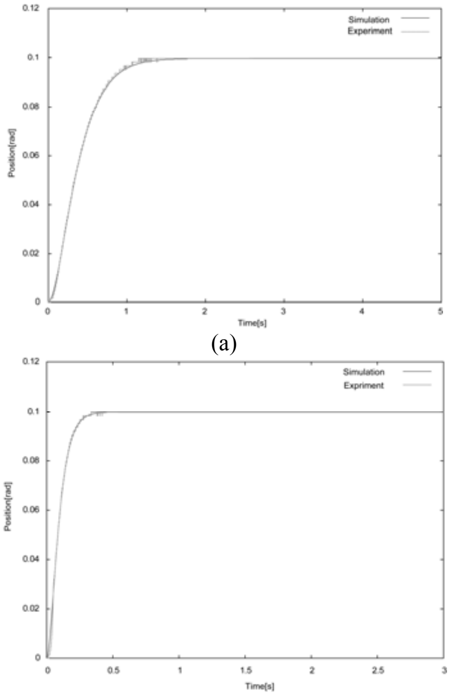


Fig. 14. Block diagram of the experimental setup.

V. Experimental Result

Experiments were performed to illustrate the performance of the proposed system. A simple peer-to-peer networked motor-control system was set up, as shown in the block diagram in Fig. 14. The system is implemented in two Altera Stratix EP1S10F780C7ES FPGAs [42]. The basic FPGA clock frequency is 33.33 MHz. The results of logic synthesis and fitting use Altera Quartus II [43]. The controller circuits take up approximately 50% of the total FPGA logic elements. The two-degrees-of-freedom controller takes 20 base clock cycles (1 clock cycle = 30 ns) to complete one control loop. This means that the controller implemented in FPGA takes 600 ns to complete one control cycle. The FPGA-based communication system takes 180 ns to transmit 1 bit in serial-communication mode. Hence, the data rate is 5.6 Mbps per channel. In combination with the communication system, a 600-ns control-sampling period is realized. The equivalent information rate in terms of the control-sampling period is 33.33 Mbps. In the experiment, a DA converter (DA7524, Analog Devices, Inc.) provides the input signal for the controlled plant. A servo amp (Yaskawa Electric Co.) and an AC servomotor (Yaskawa Electric Co.) represent the controlled plant. The servo amp works at a 12.5 kHz switching frequency. For measurement and data capture, a PC configured with RTLinux captures and measures data from the counter board (CNT24-4 (PCI), Contec Co.) based PCI bus with a 100-μs sampling period. We set different response time constants (τ = 0.05 s and τ = 0.2 s) to verify the response performance of the proposed system with the step-position references 1 rad and 0.1 rad respectively. The detail system parameters are listed in Table I. Fig. 15 shows the position-output response of the motor without noise. By comparing this with the simulation results in Fig.15, we can confirm reliable system response. The noise-generator circuit is shown in Fig. 16. Fig. 17 shows the position-output characteristics of the motor with noise over the channels. The noise frequency is 10 khz and a bit error of 0.5478% is realized. The position response of the motor is shown in Fig. 17(a). The position responses of the motor are shown in Fig. 17(b) and (c) with the noise frequency set at 30 (bit error 0.9234%) and 40 khz (bit error 1.5305%), respectively. Fig. 17(c) shows that the system response is instable and unreliable with a 1.5305% bit error. While the simulation is tolerated under 3% bit error, in the experiment, the channel noise also affects synchronization among FPGAs, something which cannot appear in the simulations. From the experimental results, we can conform that the proposed system can provide a suitable and reliable position-control performance and compensate for noise, under about 0.9% bit error, correctly.



(a)

(b)

Fig. 15. Control system response with high-speed communication links without noise. (a) τ = 0.2 s. (b) τ = 0.05 s.

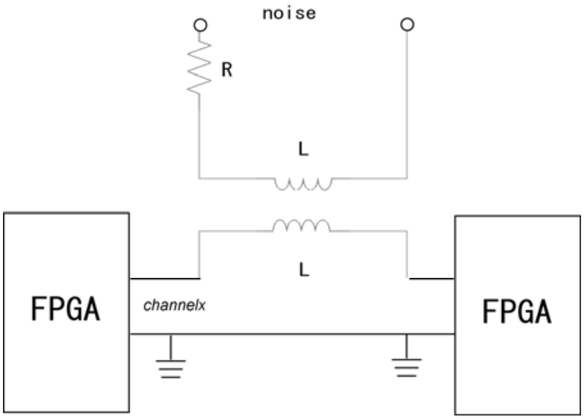


Fig. 16. Noise-generator circuit.

VI. Conclusion

We have proposed a control system with communication links that implement delta-sigma modulation to realize high-speed data transmission and data compression. The high-speed system was implemented with a 1-μs sampling period in FPGAs. The system performance was demonstrated by numerical simulation and experimental results.

VII. Appendix

In this paper, we primarily realized an angular position-control loop and communication links with 600-ns response with FPGAs. However, the circuit time constant of the current control system is 47 μs. This means that there is a constant input delay. To analyze the impact of the constant input delay we designed the two-degrees-of-freedom controller taking the delay into account. Here, the model of the controlled plant is modified by

$$P_n{}' = \frac{K_n}{(J_l s + B_n)s} \frac{1}{\tau_d s + 1}, \tag{12}$$

where τ_d is the constant input delay. We designed the command-input response of the closed-loop system considering the delay to be

$$G' = \frac{1}{\tau'^3 s^3 + 3\tau'^2 s^2 + 3\tau' s + 1}, \tag{13}$$

and the complementary sensitivity function to be

$$Q' = \frac{(2.6132\omega_c s + 1)\omega_c{}^3}{\left(s^2 + 0.7654\omega_c s + \omega_c{}^2\right)\left(s^2 + 1.6180\omega_c s + \omega_c{}^2\right)}, \tag{14}$$

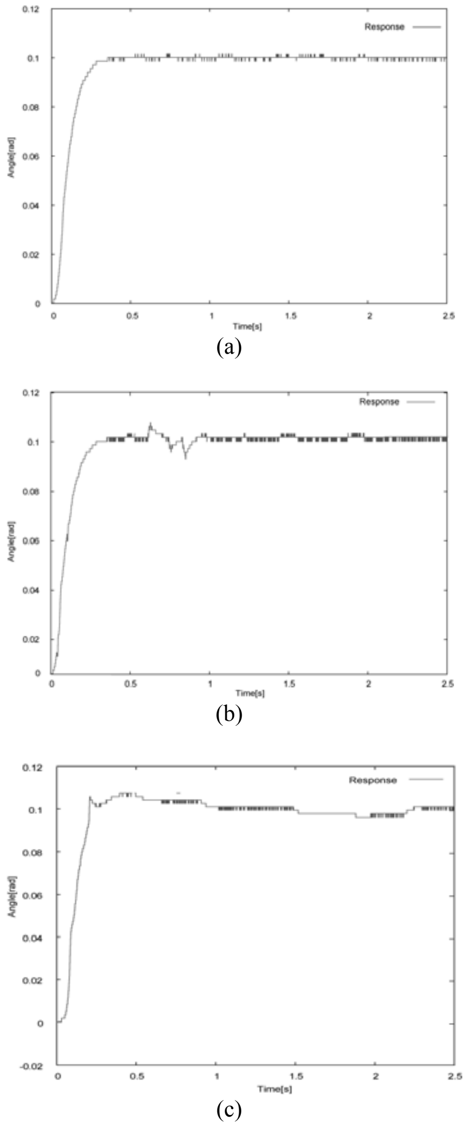
where τ' is the response-time constant. Then, the following controllers are obtained:

$$C_1{}' = \frac{G'}{P_n{}'(1 - Q')} = \frac{(B_n + J_l s)\left(s^4 + 2.6132 s^3 \omega_c + 3.41431 s^2 \omega_c{}^2 + 2.6132 s \omega_c{}^3 + \omega^4\right)(1 + \tau_d s)}{K_n s\left(s^2 + 2.6132 s\omega_c + 3.41431\omega_c{}^2\right)\left(\tau'^3 s^3 + 3\tau'^2 s^2 + 3\tau' s + 1\right)}, \tag{15}$$

$$C_2{}' = \frac{Q'}{P_n{}'(1 - Q')} = -\frac{(B_n + J_l s)(2.6132 s + \omega_c)(1 + \tau_d s)\omega_c{}^3}{K_n s\left(s^2 + 2.6132 s\omega_c + 3.41431\omega_c{}^2\right)}. \tag{16}$$

The simulation results of step response, for considering delay and without considering delay, are shown in Fig. 18, where $\tau = 70$ ms, $\tau' = 50$ ms, and $\xi = 1$. The simulation results indicate that the same response characteristic can be obtained by adjusting the response-time constant. In other words, adjusting the response time constant τ can compensate for the 47-μs time delay.

The proposed system can be expanded to a multi-node system using a network with the proposed core controller, with a 1.6-Mhz control cycle, where the MOSFET switching device realizes a PWM inverter with a 100-khz switching frequency. The multi-node system can include more than 10 nodes, where the system cycle time is satisfied with the characteristics of the switching device. This system can provide much higher speed than recent systems.



(a)

(b)

(c)

Fig. 17. Experimental result with noise over channels.
(a) 0.5478% bit error. (b) 0.9234% bit error. (c) 1.5305% bit error.

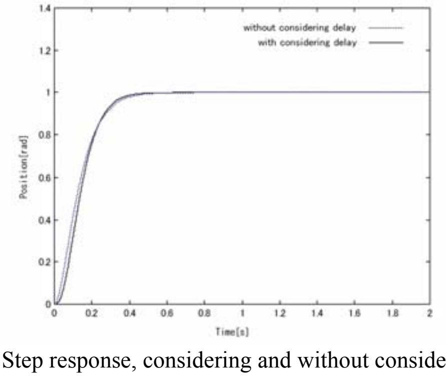


Fig. 18. Step response, considering and without considering, delay.

REFERENCES

[1] M.-Y. Chow and Y. Tipsuwan, “Gain adaptation of network DC motor controllers base on QOS variation,” *IEEE Trans. on Industrial Electronics,* vol. 50, no. 5, pp. 936-943, 2003.

[2] Y. Tipsuwan and M.-Y. Chow, “Control methodologies in networked control system,” *Control Engineering Practice,* vol. 11, pp. 1099-1111. 2003.

[3] J. A. Janet, W. J. Wiseman, R. D. Michelli, A. L. Walker, M. D. Wysochanski and R. Hamlin, “Applications of control networks in distributed robotic systems,” *in Proc. IEEE Int. Conf. on System, Men and Cybernetics*, pp.3365-3370, 1998.

[4] E. Ishii, H. Nishi and K. Onnishi, “Improvement of performances in bilateral teleopertation by using FPGA,” *IEEE Trans. on Industrial Electronics*, vol. 54, no. 4, pp. 1876-1884, 2007.

[5] R. Jurgen, “Coming from Detroit: networks on wheels,” *IEEE Spectrum,* vol. 23, no. 6, 53-59, 1986.

[6] IEC 61491, Electrical equipment of industrial machines- Serial data link for real time communication between controls and drives SERCOS, 2002-10

[7] W. Hu, G. P. Liu, and D. Rees, “Networked predictive control system with data compression,” *in Proc. IEEE int. conf. on Networking, Sensing and Control,* pp. 52-57, 2007

[8] R. W. Brockett and D. Liberzon, “Quantized feedback stabilization of linear systems,” *IEEE Trans. on Automatic Control,* vol. 45, no. 7, pp. 1279-1289, 2000.

[9] D. Liberzon, “On stabilization of linear systems with limited information,” *IEEE Trans. on Automatic Control,* vol. 48, no. 2, pp. 304-307, 2003.

[10] H. Ishii and T. Basar, “Remote control of LTI systems over networks with state quantization,” *System and Control letters,* vol. 54, no. 1, pp.15-31, 2005.

[11] N. Elia and S. K. Mitter, “Stabilization of linear systems with limited information,” *IEEE Trans. on Automatic Control,* vol. 46, no. 9, pp.1384-1400, 2001.

[12] J. Hespanha, A. Ortega and L. Vasudevan, “Towards the control of linear systems with minimum bit-rate,” *in Proc. the Int. Symposium on the Mathematical Theory of Networks and Systems,* pp.1-15, 2002.

[13] W. S. Wong and R. W. Brockett, “System with finite communication bandwidth constraints II: Stabilization with limited information feedback,” *IEEE Trans. on Automatic Control,* vol. 44, no. 5, pp.1049-1053, 1999.

[14] E. Monmasson and Y. A. Chapuis, “FPGA design methodology for industrial control systems-a review,” *IEEE Trans. on Industrial Electronics*, vol. 54, no. 4, pp. 1824-1842, 2007.

[15] E. Monmasson and Y. A. Chapuis, “Contributions of FPGAs to the control of electrical systems, *a Review,*” *IEEE Industrial Electronics Society Newsletter,* Vol. 49, No. 4, pp. 8-15, 2002.

[16] J. J. Rodriguez-Andina, M. J. Moure, M. D. Valdes, “Features, design tools, and application domains of FPGAs,” *IEEE Trans. on Industrial Electronics*, vol. 54, no. 4, pp. 1810-1823, 2007.

[17] D. H. Lee, A. Choi, J. M. Koo, J. I. Lee and B. M. Kim, “A wideband DSCDMA modem for a mobile station,” *IEEE Trans. on Consumer Electronics*, vol. 45, pp. 1259-1269, 1999.

[18] C. Dick and F. Harris, “FPGA signal processing using sigma-delta modulation,” *IEEE Signal Process. Mag.*, pp.20-35, 2000.

[19] C. Dick and F. Harris, “High-performance FPGA filters using sigma-delta modulation encoding,” *in Proc. Int. Conf. on Acoustics Speech and Signal Processing*, vol. 4, pp.2123-2126, 1999.

[20] C. Chou, S. Mohanakrishnan and J. B. Evans, “FPGA implementation of digital filters,” *in Proc. Signal Processing Applications and Technology Conf.*, pp. 80-84, 1994.

[21] E. Kappos and D.J. Kinniment, “Application-specific processor architectures for embedded control: case Studies,” *Microprocessors and Microsystems*, pp. 225-232, 1996.

[22] S. Sanchez-Solano, A. J. Cabrera, I. Baturone, F.J. Moreno-Velo, and M. Brox, “FPGA implementation of embedded fuzzy controllers for robotic applications,” *IEEE Trans. on Industrial Electronics*, vol. 54, no. 4, pp. 1937-1945, 2007.

[23] Y. A. Chapuis, L. Zhou, Y. Fukuta, Y. Mita, and H. Fujita, “FPGA-Based decentralized control of arrayed MEMS for microrobotic application,” *IEEE Trans. on Industrial Electronics*, vol. 54, no. 4, pp. 1926-1936, 2007.

[24] T. N. Chang, B. Cheng and P. Sriwilaijaroen, “Motion control firmware for high-speed robotic systems,” *IEEE Trans. on Industrial Electronics*, vol. 53, no. 5, pp. 1713-1722, 2006.

[25] M. W. Naouar, E. Monmasson, A. A. Naassani, I. Slama-Belkhodja and N. Patin, “FPGA-Based current controllers for AC machine drives—A review,” *IEEE Trans. on Industrial Electronics*, vol. 54, no. 4, pp. 1907-1925, 2007.

[26] J. C. G. Pimentel and H. Le-Huy, “A VHDL-based methodology to develop high performance servo drivers,” *in Proc. IEEE-IAS Annual Conf.*, pp. 1505-1512, 2000.

[27] T. Ide and T. Yokoyama, “A study of deadbeat control for three phase PWM inverter using FPGA based hardware controller,” *in proc. PESC’04,* Vol. 1., pp. 50-53, 2004.

[28] Y. Y. Tzou and H. J. Hsu, “FPGA realization of space-vector PWM control IC for three-phase PWM inverters,” *IEEE Trans. on Power Electronics*, vol. 12, pp.953-963, 1997.

[29] S. J. Henriksen, R. E. Betz and B. J. Cook, “Digital hardware implementation of a current controller for IM variable-speed drives,” *IEEE Trans. On Industry Applications*, vol. 35, pp.1021-1029, 1999.

[30] M. Fathallah, J. P. Chante, F. Calmon and M. H. El-husseini, “Design of an optimized IC for control algorithms of AC machines: system testing and application,” *in Proc. IEEE-IAS Annual Conf.*, pp.103-109, 2001.

[31] Y. F. Chan, M. Moallem and W. Wang, “Design and implementation of modular FPGA-Based PID controllers,” *IEEE Trans. on Industrial Electronics*, vol. 54, no. 4, pp. 1898-1906, 2007.

[32] A.-M. Lienhardt, G. Gateau, and T. A. Meynard, “Digital sliding-mode observer implementation using FPGA,” *IEEE Trans. on Industrial Electronics*, vol. 54, no. 4, pp. 1865-1875, 2007.

[33] Y. Fujimoto and A. Kawamura, “Robust servo-system based on Two-Degree-of-Freedom control with sliding Mode,” *IEEE Trans. on Industrial Electronics*, vol. 42, no. 3, pp. 272-280, 1995.

[34] T. Umeno and Y. Hori, “Robust speed control of DC servomotors using modern Two-Degrees-of-Freedom controller Design,” *IEEE Trans. on Industrial Electronics*, vol. 38, No. 5, pp.363-368, 1991.

[35] P. W. Wong, “Fully sigma-delta modulation encoded FIR filters,” *IEEE Trans. on signal process*, vol. 40, no. 6, pp. 1605-1609, 1992.

[36] J. Acero, D. Navarro, L. Barragan, I. Garde, J. Artigas, and J. Burdio, “FPGA-based power measuring for induction heating appliances using sigma–delta A/D conversion,” *IEEE Trans. on Industrial Electronics*, vol. 54, no. 4, pp. 1843–1852, 2007.

[37] S. Brown and Z. Vranesic, *Fundamentals of digital logic with VHDL design*, Tsinghua University Press, 2002.

[38] E. Ishii, H. Nishi and K. Ohnishi, “Improvement of performance in bilateral teleportation by using FPGA,” *in Proc. IEEE Int. Workshop on Advanced Motion Control*, pp. 317-322, 2006.

[39] T. Li and Y. Fujimoto, “On A control system with high-speed and real-time communication links,” *in Proc. IEEE Int. Workshop on Advanced Motion Control*, vol. 1, pp.12-17, 2006.

[40] *IEEE Standard for Binary Floating-point Arithmetic*, ANSI/IEEE Std., IEEE 754-1985, 1985.

[41] N. Shirazi, A. Walters and P. Athanas, “Quantitative analysis of floating point arithmetic on FPGA based custom computing machines,” *in Proc. IEEE Symposium on FPGAs for Custom Computing Machines,* pp.155-162, 1995.

[42] Altera, Inc., *Stratix device handbook*, San Jose, CA, 2005.

[43] Altera, Inc., *Quartus II version 5.0 handbook,* San Jose, CA, 2005.

Tianjian Li received B.S. degree from Northeastern University, Shenyang, China in 2002 and M.S. degree from the Department of Electrical and Computer Engineering, Yokohama National University, Japan, in 2006, where he is currently working toward the Ph.D. degree

His research interests include networked control systems, motion control, and field-programmable gate-array design.

Yasutaka Fujimoto (S’93-M’98) received B.E., M.E., and Ph.D. degrees in electrical and computer engineering from Yokohama National University, Yokohama, Japan, in 1993, 1995, and 1998, respectively.

In 1998, he was with the Department of Electrical Engineering, Keio University, Yokohama. Since 1999, he has been with the Department of Electrical and Computer Engineering, Yokohama National University, where he is currently an Associate Professor. His research interests include actuators, robotics, manufacturing automation, discrete event systems, and motion control.

Dr. Fujimoto is a member of IEE of Japan, Robotics Society of Japan.