Doctoral Thesis

High-Speed Control System Using Sigma-Delta Modulation and Application to Servo Systems

(ΣΔ変調を用いた高速制御用通信システムの開発およびサーボシステムへの応用)

Li Tianjian

Department of Electrical and Computer Engineering
Graduate School of Engineering
Yokohama National University

Supervisor Professor: **Yasutaka Fujimoto**

March 2009

ACKNOWLEDGMENTS

First of all, I would like to thank my supervisor Professor Yasutaka Fujimoto for his invaluable guidance, encouragement, academic stimulus and generous help. From Professor Yasutaka Fujimoto, I learnt not only the knowledge, but also the rigorous scientific approach and the dedicating spirit for work. In addition, he has infused me with enthusiasms for researching and open mind for different opinions. Without his supervision, I would not have completed this work.

I would like to thank Professor Atsuo Kawamura and Hiroshi Fujimoto who gave me advices and guidance to complete this work. I also would like to thank the previous and current members of Yasutaka Fujimoto laboratory. Discussion with my colleagues always stimulated me.

Finally, I would like to thank all my family members for their encouragement and support.

Li Tianjian

Yokohama National University

March 2009

ABSTRACT

Networked control systems are becoming increasingly important as industries focus on distributed computing in automation. For the industrial requirement, the system must combine safety with real-time demand. In real-time networked control systems, the real-time requirement is much more difficult to obtain for motion control systems than with other applications. For effective high performance motion control, demand data and feedback data must be transmitted synchronously, with very short cycle times and low latency. Any transmission delay represents a phase delay in the control system, which limits the achievable gain and the effective response time of a machine. For the existing networked control system, it is difficult to shorten the sampling period because of the limited data rate. Hence, we proposed a high speed networked control system using sigma-delta modulation for the servo system

In this paper, sigma-delta modulation is adopted to realize high speed data transmission and data compression. In this way, it can meet the requirements of the control and communication system for high speed sampling and real time. Next, the novel noise compensation scheme is presented. An FPGA solution is provided to realize the proposed system. Finally, the proposed high speed system is implemented with a 0.6-μs sampling period position control system using a servo amplifier; a 10-μs cycle time networked control system is realized in the experiments. In the current control system, the 100 kHz PWM inverter was designed. The total execution time of the controller and communication was 1.82 μs. The system performance was demonstrated by

numerical experimental results. The abovementioned features of the proposed system allow the achievement of a much higher real-time performance than the existing real-time networked control system

Contents

Figure List

Table List

Chapter 1
1 Introduction

1.1. Background of the research

Continuous improvements in automation and the growth of production scales in manufacturing have resulted in the increased use of communication technology in control systems. Networked control systems are control systems comprised of the system to be controlled and of actuators, sensors and controllers, whose operation is coordinated via a communication network, as shown in Figure 1-1.

Control systems with spatially distributed components have existed for several decades. In the past, components of such systems were connected via hardwired connections, and the systems were designed to bring all the information from the sensors to a central location. Today, improved technology allows for low cost processing power at remote locations via microprocessors and information that can be transmitted reliably via digital networks. These technology driven changes are fueled by the high costs of wiring and the difficulty in introducing additional components into systems as requirements for their use change. Consequently, networked control systems have been increasingly applied in a wide range of domains, including mobile sensor networks [1], remote surgery [2], automated highway systems [3], robotic systems [4], bilateral teleoperation systems [5] and networks in automobiles [6], aircraft and manufacturing plants.

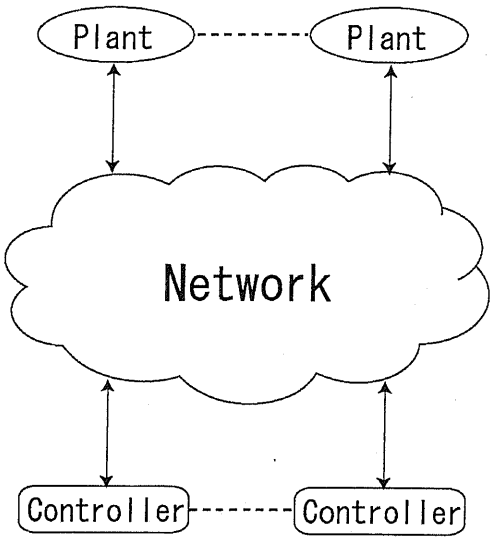


Figure 1-1: Networked control systems architecture

Networked control systems research lies primarily at the intersection of three research areas: 1) control systems, 2) communication networks and information theory and 3) computer science. Recently, networked control systems have greatly benefited from developments in information theory and computer science.

1.2. Motivations and outline of the paper

Since the advent of fieldbus [56] system, such as CAN [7], PROFIBUS [58], P-Net [59], networked control systems have undergone large improvements and received a great deal of attention. Along with developments in communication technology, the field of networked control systems has been progressing quickly. For example, networked control systems are commonly used in motion systems, in which the synchronization of different nodes is required. When the movements of various axes in a machine tool or a robot must be coordinated to follow a precise path, this requires the support of a network. In such a system, this is a tough requirement to meet. For instance, a robot controller based on the time triggered approach will read the positions of the joints to calculate the absolute position of a robot's extremity. If the positions are not sampled at the same time, then the calculation will be wrong and the controller will determine the wrong position for the robot.

Currently, several networked control system solutions have been proposed to meet this requirement. Buses used for distributed motion control include Ethernet, IEEE 1394, RS 232/485 and so forth [9]. In particular, the use of Ethernet is highly appealing, as Ethernet modules can operate at a distance as much as 85 meters from the host controller. In addition, Ethernet-based motion control and measurement devices allow complex automated measurement with the separate nodes connected by a single master controller, while standard Ethernet is incapable of meeting the requirements of real time. Different propositions exist for modifications of Ethernet technology. In principle, there are different approaches to building a real-time Ethernet solution [8], [55]. One is to maintain the TCP/UDP/IP protocols unchanged and concentrate all real-time modification in the top layer. This structure is the basis for Ethernet/IP, Modbus/TCP and so forth. Others the TCP/UDP/IP protocols are not adopted, which can be increasingly achieved as real time is performed. SERCOS III [10], [60] (its own standard

IEC 61491), Powerlink (defined by Bernecker and Rainer and supported by the Ethernet Powerlink Standardization Group), PROFINET (defined by manufacturers and supported by PROFIBUS International) MECHATROLINK-III (defined by Yaskawa Electric) and others are based on this approach. SERCOS III communication consists of two different logical communication channels: the real-time channel and the non-real-time channel. The sequence of transmitting synchronization, RT data telegrams and IP telegrams is repeated every communication cycle. The values for a communication cycle are 31.25 μs, 62.5 μs, 125 μs, 250 μs and integer multiples of 250 up to 65000 μs.

Modern control systems take from each axis multiple demand inputs and multiple feedback inputs, such as actual torque and velocity, to compute new target data for each motor. For effective high performance control, demand data and feedback data must be transmitted synchronously, with very short cycle times and low latency. Any delay in transmission represents a phase delay in the control system, which limits the achievable gain and effective response time of a machine. Distributed applications in control environments require tight synchronization in order to guarantee the delivery of control messages within defined message cycle times (typical times are given in Table 1-I [49], [50]). Cycle time includes the time taken to acquire feedback data, perform matrix computation and transmit new target data. The key technological elements that are required to achieve this are a fast synchronous network and a fast processing engine. Traditional Ethernet and fieldbus systems cannot meet cycle time requirements below a few microseconds, as shown in Table 1-II. The fastest existing system, SERCOS III, features a 31.25 μs cycle time for eight nodes at a communication data rate of 100-Mb/s. In addition, shortening the sampling period is difficult because of the limited data rate. Hence, several researchers aim at reducing the amount of information transmitted over the communication channels with data compression methods [11]. Some recent papers have focused on the

quantization problem caused by the limited channel rate [12]–[17]. Other researchers have focused on developing a new control theory under communications constraints [16], [17], and yet other researchers have focused their efforts on implementing low data rate systems for control purposes [14]. The problem of linear system stabilization under sampled encoded measurements has been studied [12], [13], based on a dynamically adjustable "zoom-in/zoom-out" quantizer. More recently, time invariant memoryless logarithmic quantizers have also been proposed [14], of which the networked control systems employ control sampling periods of microseconds or milliseconds. However, for robotic systems or high speed, high precision multi axis drive systems, a greater number of drives and a shorter cycle time are required to assure reliability and provide the desired performance.

In this paper, the proposed networked control system focuses on the motion control system and aims to realize a much shorter cycle time than that of the existing system. For the servo control system, the existing networked control system provides the solutions on the master controller to the servo amplifier level. In the proposed system, the AC servomotor minor current control loop is also considered. The communication channels are all real time with a bandwidth of 50 MHz working serially. The information to be transmitted is compressed using sigma-delta modulation (SDM), which can reduce the utilization ratio of bandwidth. With this communication protocol, the system can achieve much higher real-time performance than the existing systems. Hence, the proposed system works at a cycle time of 600 ns, without considering the minor loop of the AC servomotor, and at a 10 μs cycle time with a PWM rate of 100 kHz. This is a much higher rate than the existing systems. On the other hand, existing networked control systems often use cyclic redundancy codes (CRC) or checksums and parity codes to detect noise over the communication channels and compensate for it with the ACK method. In the proposed system, noise compensation is performed by control law, which can ignore the time required to use the

ACK method and reduces the traffic of the communication channels.

Table 1-I Typical Cycle Times for Control Applications

Control Application	Typical Cycle Time
Low speed sensors (e.g. pressure, temperature)	Tens of milliseconds
Common electro-mechanical devices (e.g. relays, breakers, solenoids)	milliseconds
Motion control (e.g. robotics)	Hundreds of microseconds
Precise motion control (e.g. high speed packaging, printing, robotics)	Tens of microseconds
High speed electrical devices (e.g. synchrophasor measurements)	Microseconds
Electronic ranging (e.g. fault detection, triangulation)	Microseconds

Table 1-II Comparison of the existing networked control system with proposed system

	CAN	SERCOS III	Powerlink	Profinet-IRT	Mechatrolink III	Proposed system
Min.Cycle Time	1ms	8 axes @ 32.5us	200us	200us	4 axes @ 125us	10us
Bandwidth	1Mbps	100Mbps	100Mbps	100Mbps	100Mbps	50Mbps

Since the invention of the field programmable gate array (FPGA) by Xilinx in 1984, FPGA technology has undergone tremendous development and continues to gain momentum. The worldwide FPGA market is expected to grow from $1.9 billion in 2005 to $2.75 billion by 2010. FPGAs have gone from being simple glue logic chips to actually replacing custom application specific integrated circuits (ASICs) and processors for signal processing and control applications [18], [19], [20], [36].

At the highest level, FPGAs are reprogrammable silicon chips. An FPGA is defined as a matrix of configurable logic blocks, linked to one other by an entirely reprogrammable interconnection network. The memory cells control both the logic blocks and the connections so that the component can fulfill the required application specifications. The adoption of FPGAs across all industries is driven by the fact that FPGAs combine the best parts of ASICs and processor-based systems. For example, FPGAs provide hardware-timed speed and reliability, but do not require high volumes to justify the large upfront expense of custom ASIC design. In addition, while reprogrammable silicon has the same flexibility of software running on a processor-based system, it is not limited by the number of processing cores available. Unlike processors, FPGAs are truly parallel in nature. Therefore, different processing operations need not compete for the same resources. Taking advantage of hardware parallelism, the computing power of FPGAs exceed that of digital signal processors (DSPs) by breaking the paradigm of sequential execution and accomplishing more per clock cycle. FPGAs offer flexibility and rapid prototyping capabilities in the face of increased time-to-market concerns. They allow the testing of an idea or concept and its verification in hardware, without going through the long fabrication process of custom ASIC design. Furthermore, FPGAs are not easily duplicable because of their specific architecture. Due to these benefits, FPGAs are used in a variety of industries and applications, such as wired and wireless communications [21], signal processing [22], [23], [24], embedded control systems [25]

and robotics [26], [27], [28]. Moreover, FPGAs have already been used with success in many different electric system applications instead of solutions involving DSPs. Naouar et al. present the interest in implementing digital controllers for AC machine drives using FPGA [29]. ON-OFF current controllers, PI current controllers and predictive current controllers have been designed and implemented on FPGAs, whose calculation time are 2.64 μs, 2.64 μs and 4.52 μs respectively. The quality of the regulated current is significantly improved, mainly due to a very important reduction in execution time delay, which is only a few microseconds. This time reduction derives directly from the possibility offered by FPGAs to design very powerful dedicated architectures. Yokoyama et al. present a new method for the real-time digital feedback control of a three phase PWM inverter, in which a deadbeat control is realized with the use of an FPGA [31]. Simulations and experimental results verify that the FPGA-based deadbeat controller can provide a good performance with the total calculation time of 1.15 μs. Similarly, S. J. Henriksen et al. propose an FPGA solution for an induction machine digital current controller [32]. In [34], the authors present a sensorless neural network-based induction motor control scheme, developed by following a holistic approach to electronic system modeling and controller design, and give the practical example with an FPGA implementation. The FPGA solutions for motion control have also been widely used. In [5], the authors describe a high performance bilateral teleoperation FPGA-based controller with a disturbance observer. The implementation of FPGA resulted in the significant reducing of the sampling period of the controller and the consequent widening of the bandwidth of the force estimation, whose sampling period is 10 μs.

In this paper, an entire FPGA solution is provided for not only the communication system but also the position controller and the current controller. This paper shows the development of a high speed real-time control and communication system, with communication channels connecting the controller and the controlled plant. The communication system uses sigma-delta

modulation to compress the transmitted data. The system gains communication efficiency through this modulation method and successfully compensates for noise over the communication channels by setting the compensation channel and the designed servo controller.

In chapter two, a novel high speed real-time control and communication system is proposed. The structure of the system, including the remote control system and the communication system, is presented. In the communication system, the modified sigma-delta modulation is employed to coding source between the both sides of the proposed system.

In chapter three, the application to the servomotor system is presented. The controlled plant is treated as a servomotor, which is driven by a servo amplifier. Thus, the voltage of information, which is the reference of the servomotor generated by the remote controller, is transmitted over the communication channels. For the noise over the communication channels, the designed controller can compensate for the noise over the reference channel (channel1 in Figure 2-1), but the noise over the feedback channel (channel2 in Figure 2-1) cannot be compensated for because this noise is unobservable. To compensate for the noise over channel2, we designed the compensation channel (channel3 in Figure 3-7). The scheme of channel3 is presented in this chapter. The simulation results are also given in this chapter.

In chapter four, an extended system is proposed to compensate for the sampling speed limitation of the servo amplifier. This system considers the minor current control loop of the servomotor. The simulation results are shown, and the novel noise compensation method is also proposed in this chapter.

In chapter five, the error distribution for the proposed communication system is presented. The difference in error distribution causing the coding scheme of the sigma-delta modulation is also discussed in this chapter. For the safety level in the industrial environment, a discussion of the proposed noise compensation scheme is also included.

In chapter six, the FPGA implementation for the proposed networked control system is developed, for its achievement of high speed and high performance levels. Here, the architecture of the circuit based on an FPGA is described. The FPGA solution for the proposed system is divided into two parts: one is the communication system solution, and the other is the control system solution. In the control system solution, the position controller and the current controller are implemented in floating point.

In chapter seven, the experiments for the motor drive as a servo amplifier are performed to illustrate the performance of the proposed system. The experimental setup is described, and the results for the noise are also given in this chapter. Through this chapter we can confirm the validity of the proposed system. For the FPGA solution system, the control cycle is 600 ns, which is significantly higher than that of the existing networked control system.

Chapter eight features the experiments using a current control loop with the FPGA solution, which were performed to illustrate the proposed system's performance. The experimental setup is described, and a 100 kHz PWM is realized, with the controller execution time of 1.82 μs. The cycle time of the proposed networked control system is 10 μs. The position response results of the motor are also given in this chapter.

In chapter nine, the paper is concluded with a summary.

Chapter 2

2 Configuration of the Real-time Control System with High-speed communication links

2.1. System overview

The structure of the control system with communication channels is given in Figure 2-1. In this system, the controller and the plant are connected via communication channels. The channels transport digitized information from the controller to the actuator node and also convey sensor information back to the controller. As the solution for the bandwidth limitations of digital communication channels, the sigma-delta modulation is utilized to reduce communication traffic. The controlled plant is represented by a servomotor. Compensation for noise for the communication channels has also been included in the system's design. On the other hand, as the proposed system is to be applied in short haul connections such as a robotics system, the transmission delays are neglected.

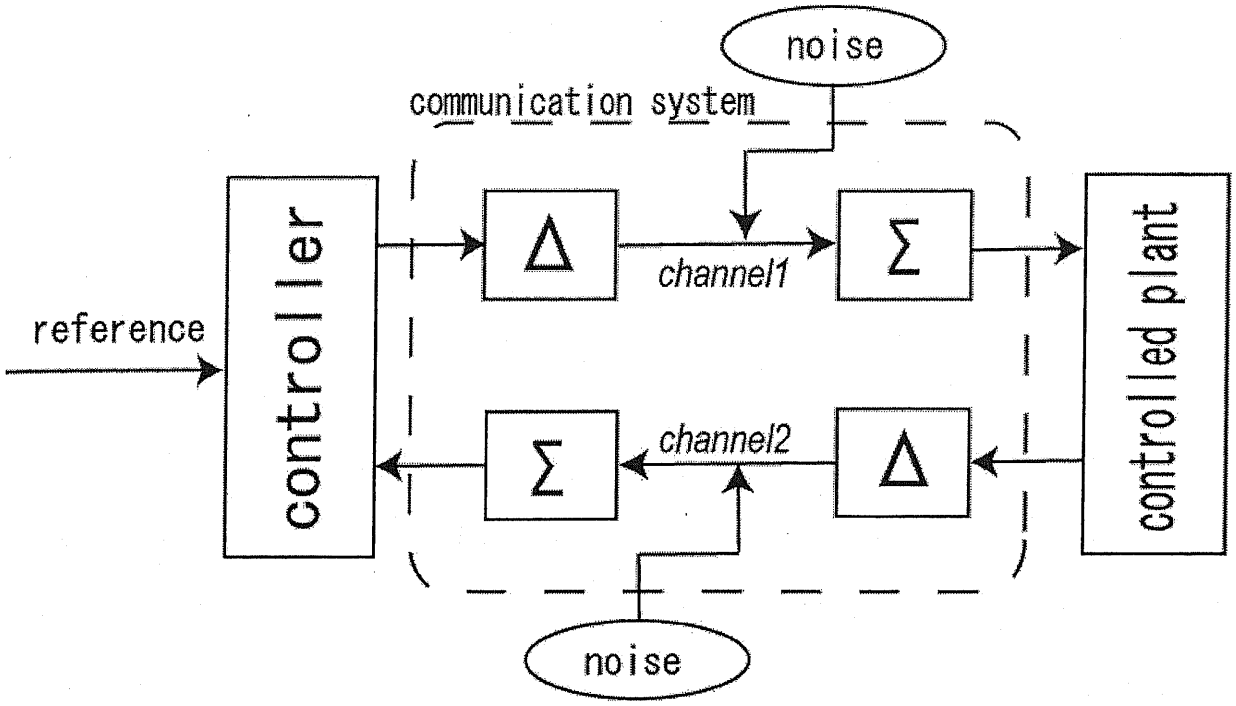


Figure 2-1: Configuration of the control system with communication channels

2.2. Communication system design

This problem is relevant in the area of networked control systems, where several applications call for data compression algorithms aimed at reducing the amount of information that may be transmitted through communication channels and therefore permit a better resource allocation and an improvement of the permissible closed loop system data rate.

In the proposed system, we utilize sigma-delta modulation to compress data in the communication system. sigma-delta 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 [22], [40], [41]. Using sigma-delta modulation, various signal-processing application circuits can be designed to achieve high bit resolution with low hardware cost. Sigma-delta modulation can convert the large

bit-width data format to a one-bit format, reducing the traffic in the communication channels. The sigma-delta modulated data stream is low-pass filtered to reduce the quantization noise; thus, it meets the requirements of the control/communication system for high speed sampling and real-time operation. The fundamental difference between the proposed sigma-delta modulation and the classic version is a three level (1, 0, -1), two-bit dynamic quantizer instead of the conventional two levels (1,-1), which codes by one bit.

For the motor control system, the information to be transmitted includes position, velocity, torque, current, voltage and so forth, all of which are unbroken curves and are rated. For the position channel and current channel, if Δ_θ and Δ_I meet the following conditions, the band will be satisfied.

$$\Delta_\theta \geq \frac{2\pi \cdot n_{\max} \cdot T}{60}$$
$$\Delta_I \geq \frac{I_{\max} \cdot T}{T_e} \qquad (2.1)$$

Where T is the sampling period, $n_{\max}$ is the motor rated maximum speed, $I_{\max}$ is the motor rated maximum instantaneous current and T_e is the motor electric time constant. Therefore, the sigma-delta modulation can be applied to a networked control system for motor control. Additionally, using sigma-delta modulation allows the sampling period to be increased to safe the sigma-delta modulation band, and the values of the motor are rated. On the other hand, we designed the additional channel to compensate for the noise, for which the controller cannot compensate. The channel not only compensates the noise over the channel but also satisfies the band of sigma-delta modulation.

2.2.1. Delta modulation

Delta modulation is the simplest form of differential pulse-code modulation (DPCM) where the difference between successive samples is encoded into n-bit data streams. In delta modulation, the transmitted data is reduced to a one-bit data stream. In the proposed system, the delta modulation is modified to a three level output with a two-bit data stream. It 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 Figure 2-2, where the equation is

$$u_q(k) = \begin{cases} 1 & if \quad u(k) - u(k-1) \geq \Delta u \\ 0 & if \quad |u(k) - u(k-1)| < \Delta u \\ -1 & if \quad u(k) - u(k-1) \leq -\Delta u \end{cases} \tag{2.2}$$

where Δu is a quantum unit of the discrete system model.

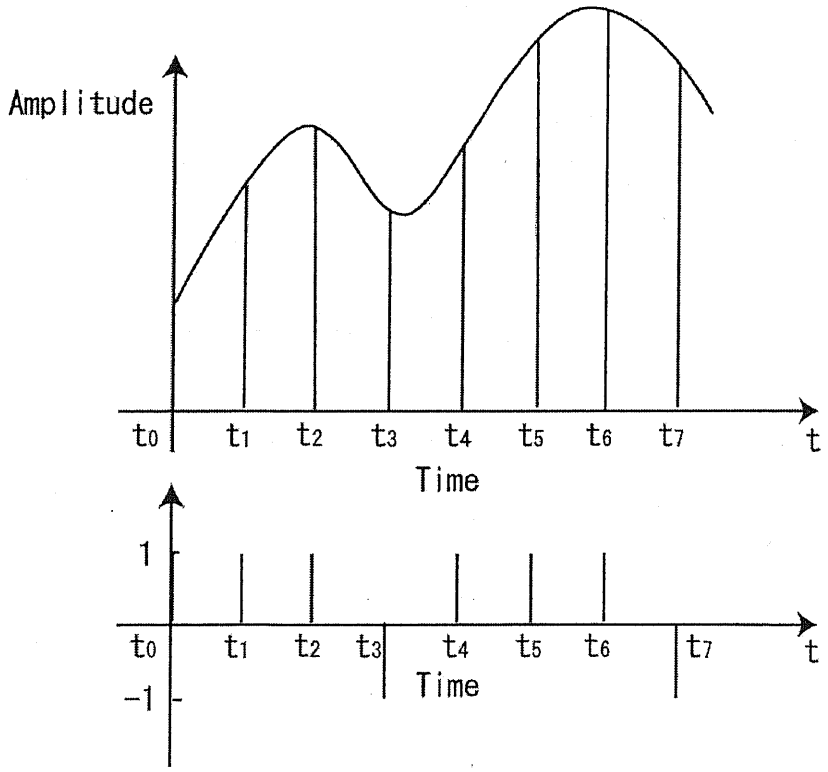


Figure 2-2: Function of delta modulation.

2.2.2. Sigma modulation

Sigma modulation is the integrator that adds and subtracts the value from the output of delta modulation. The function is shown in Figure 2-3, where the equation is

$$\hat{u}(k) = \hat{u}(k-1) + \Delta u \cdot u_q(k). \tag{2.3}$$

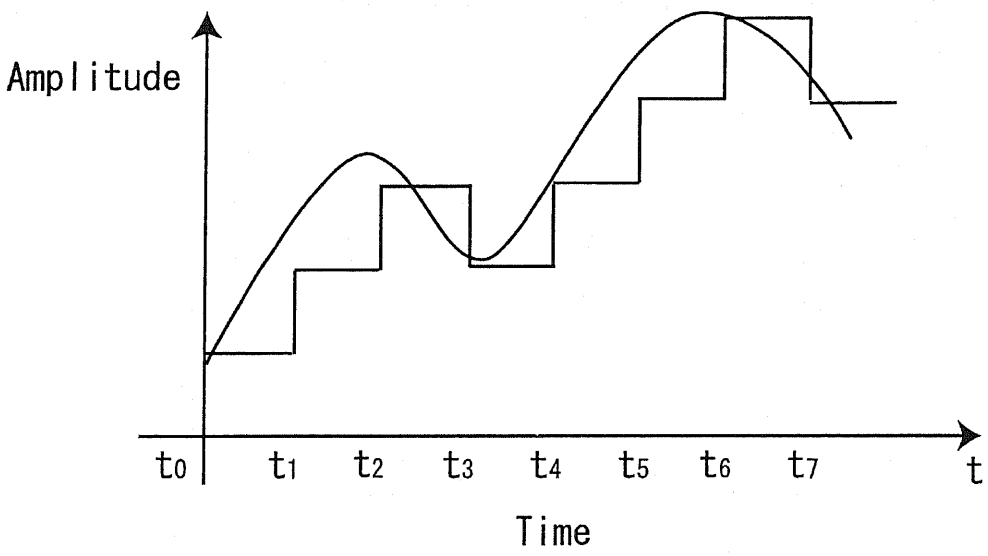


Figure 2-3: Function of sigma modulation.

2.3. Stability analysis for control system with sigma-delta modulation communication links

We consider the systems with a linear feedback of the form

$$\begin{aligned} \dot{x} &= Ax + Bu \\ u &= -K\hat{x} \end{aligned} \tag{2.4}$$

where A and B are controllable matrices, $u(t)$ is the input, K is the feedback gain and $\hat{x}$ is the estimated value of x as obtained from the sigma-delta modulation process. Considering the sigma-delta modulation in the feedback system, the error equations are

$$\begin{aligned} \dot{x} &= A_c x + BK\tilde{x} \\ \dot{\tilde{x}} &= A_c c + BK\tilde{x} - \Delta_u \operatorname{sgn}(\dot{x}) \end{aligned} \tag{2.5}$$

where $\tilde{x} = x - \hat{x}$ and $A_c = A - BK$. $\operatorname{sgn}(\dot{x})$ is the continuous time version of the sigma-delta modulation with the value (1, 0, -1). Assume that Δ_u is constant and has the following form

$$\Delta_u = \Delta_0 I_{n\times n}$$

where $\Delta_0 \geq c \cdot \|\varsigma(0)\|$. Here, further definitions and notations are given.

$$\varsigma = (x^T, \tilde{x}^T)^T,$$

$$c > \max(c_5, c_5 c_6) > 0 \;;\; c_1 = \|2PBK\| \;;\; c_2 = \|2A_c\| \;;$$

$$c_3 = \|2BK\| \;;\; c_4 = c_2 + c_3 \;;$$

$$c_5 = c_3 + \frac{c_4^2}{4q} \;;\; c_6 = \sqrt{\frac{\lambda_{\max P} + 1}{\lambda_{\min P} + 1}}$$

where $\lambda_{\min} P$ and $\lambda_{\max} P$ are the maximum and minimum eigenvalues of $P = P^T > 0$, and $q = \lambda_{\min} Q$ is the minimum eigenvalue of $Q > 0$, solutions of $PA_c + A_c^T P = -Q$.

Consider the quadratic Lyapunov function $V = x^T Px + \tilde{x}^T \tilde{x}$, and the $\dot{V}$ is given as

$$\begin{aligned} &\dot{V} \le -q\|x\|^2 + c_1\|x\|\|\tilde{x}\| + c_2\|x\|\|\tilde{x}\| + c_3\|\tilde{x}\|^2 - \Delta_0|\tilde{x}| \le \\ &-q\|x\|^2 + c_4\|x\|\|\tilde{x}\| + c_3\|\tilde{x}\|^2 - \Delta_0\|\tilde{x}\| \le \\ &-(\|x\|,\|\tilde{x}\|)\begin{pmatrix} q & -\frac{c_4}{2} \\ -\frac{c_4}{2} & \frac{\Delta_0}{\|\tilde{x}\|} - c_3 \end{pmatrix}\begin{pmatrix} \|x\| \\ \|\tilde{x}\| \end{pmatrix} \end{aligned} \tag{2.6}$$

If the value of Δ_0, such that $q(\frac{\Delta_0}{\|\tilde{x}\|} - c_3) > \frac{c_4^2}{4}$, we can see that $\dot{V}$ is negative. Nevertheless, this condition by itself does not define the domain of attraction. For this, C. Canudas-de-Wit et al give the proof in [53]. From [53] the relation $\Delta_0 > c_5\|\varsigma\|$ is obtained.

For the observer system shown in Figure 2-4, the state function can be given as

$$\begin{aligned} &\dot{x} = Ax + Bu \\ &y = Cx \\ &u = F\hat{x} + C_q x_q \\ &\dot{\hat{x}} = A\hat{x} + Bu + H(\hat{y} - z) \\ &\dot{x}_q = A_q x_q + B_q(\hat{y} - z) \\ &e = \hat{x} - x \\ &\dot{z} = \Delta\,\mathrm{sgn}(\dot{y}) \end{aligned} \tag{2.7}$$

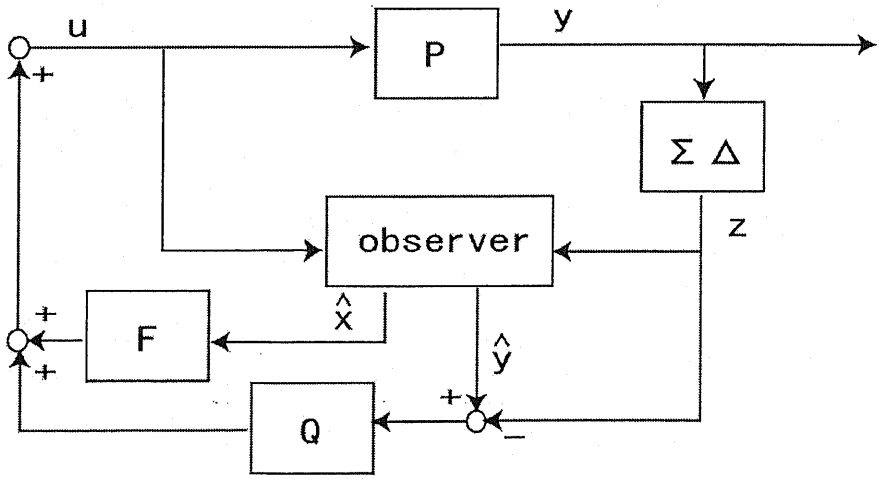


Figure 2-4: Block diagram of observer with sigma-delta modulation

where $x \in \Re^n$, $\hat{x} \in \Re^n$, $x_q \in \Re^m$, $A \in \Re^{n \times n}$, $B \in \Re^{n \times 1}$, $C \in \Re^{1 \times n}$, $A_q \in \Re^{m \times m}$, $B_q \in \Re^{m \times 1}$, $C_q \in \Re^{1 \times m}$, and A_q is stable. z is the output of sigma-delta modulation. F and H are the feedback gain to guarantee $A_F = A + BF$ and $A_H = A + HC$ are strictly stable. $\hat{x}, \hat{y}$ are the state of estimate by the observer. From equation 2.7 the error equation is

$$\frac{d}{dt}\begin{bmatrix} x \\ e \\ x_q \\ z \end{bmatrix} = \begin{bmatrix} A_F & BF & BC_q & 0 \\ HC & A_H & 0 & -H \\ B_qC & B_qC & A_q & -B_q \\ 0 & 0 & 0 & 0 \end{bmatrix}\begin{bmatrix} x \\ e \\ x_q \\ z \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 0 \\ \Delta \mathrm{sgn}(\dot{y}) \end{bmatrix} \tag{2.8}$$

Here, further definitions and notations are given. $\lambda_{\min Q1}, \lambda_{\min Q2}, \lambda_{\min Q3}$ are the minimum eigenvalue of $Q_1 > 0, Q_2 > 0, Q_3 > 0$, solutions of $A_F{}^T P_1 + P_1 A_F{}^T = -Q_1$, $A_H{}^T P_2 + P_2 A_H{}^T = -Q_2$, $A_q{}^T P_3 + P_3 A_q{}^T = -Q_3$, where $P_1 = P_1^T > 0, P_2 = P_2{}^T > 0, P_3 = P_3{}^T > 0$.

Using the same method in [53], consider the lyapunov function $V = x^T P_1 x + e^T P_2 e + x_q{}^T P_3 x_q + z^T z$, Evaluation $\dot{V}$ along solutions of equation 2.8 gives

$$\dot{V} \le -\begin{bmatrix} \|x\| \\ \|e\| \\ \|x_q\| \\ \|z\| \end{bmatrix}^T \begin{bmatrix} \lambda_{\min Q1} & -\left(\|P_1BF\| + \|P_2HC\|\right) & -\left(\|P_3B_qC\| + \|P_1BC_q\|\right) & 0 \\ -\left(\|P_1BF\| + \|P_2HC\|\right) & \lambda_{\min Q2} & -\|P_3B_qC\| & \|P_2H\| \\ -\left(\|P_3B_qC\| + \|P_1BC_q\|\right) & -\|P_3B_qC\| & \lambda_{\min Q3} & \|P_3B_q\| \\ 0 & \|P_2H\| & \|P_3B_q\| & \frac{2\Delta}{\|z\|} \end{bmatrix} \begin{bmatrix} \|x\| \\ \|e\| \\ \|x_q\| \\ \|z\| \end{bmatrix} \tag{2.9}$$

From here we can see that $\dot{V}$ is negative as long as the matrix in the equality above is positive define. If the value of Δ, such that

$$\Delta > \frac{\|P_3B_q\|\left(\lambda_{\min Q1}\lambda_{\min Q2}\|P_3B_q\| + \lambda_{\min Q1}\|P_2H\|E - G^2\|P_3B_q\| + G\|P_2H\|L\right) - \|P_2H\|\left(L^2\|P_2H\| - \lambda_{\min Q1}E\|P_3B_q\| - \lambda_{\min Q1}\lambda_{\min Q3}\|P_2H\| - LG\|P_3B_q\|\right)}{\lambda_{\min Q1}\lambda_{\min Q2}\lambda_{\min Q3} - \left(\lambda_{\min Q1}E^2 + 2GEL + \lambda_{\min Q3}G^2 + \lambda_{\min Q2}L^2\right)}\|z_0\| \tag{2.10}$$

where $G = \|P_1BF\| + \|P_2HC\|$, $E = \|P_3B_qC\|$ and $L = \|P_3B_qC\| + \|P_1BC_q\|$ are defined. $\|z_0\|$ is the initial conditions. Here, the relations $\lambda_{\min Q1}\lambda_{\min Q2}\lambda_{\min Q3} - \left(\lambda_{\min Q1}G^2 + 2GEL + \lambda_{\min Q3}G^2 + \lambda_{\min Q2}L^2\right) > 0$ and $\lambda_{\min Q1}\lambda_{\min Q2} - G^2 > 0$ need to be met.

Chapter 3

3 Application to Servo Motor Control without Considering the Current Control Loop

In the previous section, the system structure and the sigma-delta modulation are introduced. For the proposed networked control system, this paper focus on the motion control with the real-time communication links. It is desired to provide a high-performance solution to robotics system. For almost robotics systems, the servo motor is adopted to constitute actuators. So the robotics system can be considered as a multiaxial servo system. Hence, in the proposed system the servo motor is treated as the controlled plant, and the controller is designed according to the servo system to provide a high performance position control. A type 1 servo controller can compensate for step disturbances on the input. In this paper a two-degrees-of-freedom controller is employed to realize the developed control system. The control system configuration is shown in Figure 3-1. Figure 3-2 shows the configuration of a two-degrees-of-freedom control system with a communication system using sigma-delta modulation in Z-transform domain. The transfer function can be simply given as

$$u = \begin{bmatrix} C_1 & C_2 \end{bmatrix} \begin{bmatrix} r \\ y \end{bmatrix} \tag{3.1}$$

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

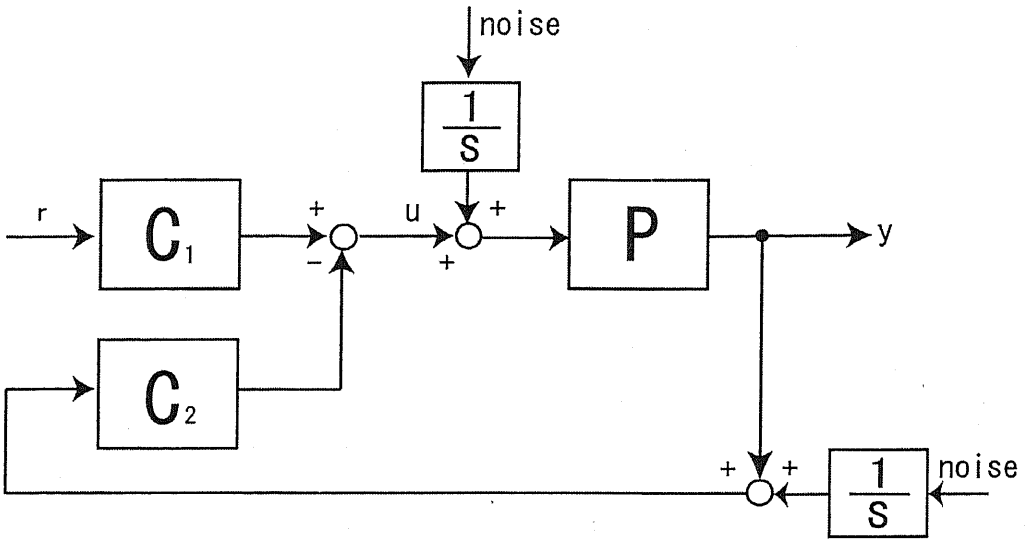


Figure 3-1: Configuration of the two-degrees-of-freedom control system

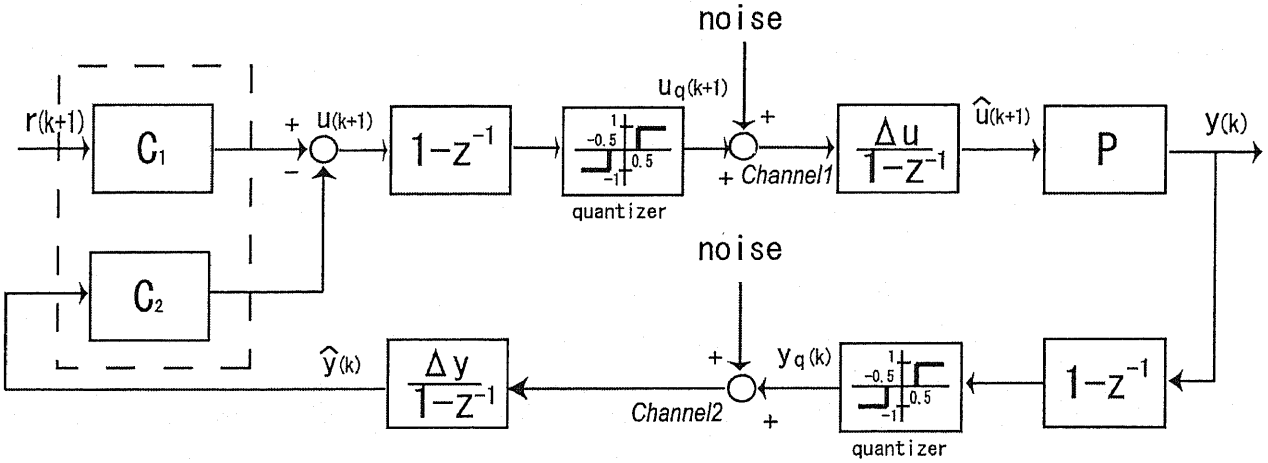


Figure 3-2: Configuration of the control system with sigma-delta modulation in Z-transform.

3.1. Position controller

A servo motor is treated as the controlled plant to verify the developed system by numerical simulation. The nominal model of the controlled plant is expressed as

$$P_n = \frac{K_n}{(J_l s + B_n)s}, \tag{3.2}$$

where B_n and K_n are

$$B_n = \frac{K_e K_T + R_a D_l}{R_a}, \tag{3.3}$$

$$K_n = \frac{K_T}{R_a}, \tag{3.4}$$

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. The command-input response of the closed-loop system is

$$G = \frac{1}{\tau^2 s^2 + 2\xi\tau s + 1} \tag{3.5}$$

and the complementary sensitivity function to be

$$Q = \frac{(2\omega_c s + 1)\omega_c{}^2}{(s^2 + \omega_c s + \omega_c{}^2)(s + \omega_c)} \tag{3.6}$$

where τ is the response-time constant, and ω_c is the damped frequency. This provides the following controllers [38], [39]:

$$C_1 = \frac{G}{P_n(1-Q)} = \frac{(B_n + J_l s)(s^3 + 2s^2\omega_c + 2s\omega_c{}^2 + \omega_c{}^3)}{K_n s(s + 2\omega_c)(1 + s^2\tau^2 + 2s\tau\xi)}, \tag{3.7}$$

$$C_2 = \frac{Q}{P_n(1-Q)} = -\frac{(B_n + J_l s)(2s + \omega_c)\omega_c{}^2}{K_n s(s + 2\omega_c)}, \tag{3.8}$$

The specifications of the motor and the parameters of the controller are shown in Table 3-I. The input-channel quantization value Δu_{in}, and the output channel quantization value, Δu_{out}, are set to $10.0/256[V]$ and $0.5\pi/4096[rad]$ respectively. The simulation is written in C. To

verify the resistance to noise, the noise is set to be about a 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 a level of approximately 3% because of the PWM switching noise. This is a realistic noise level for real conditions. The two-degrees-of-freedom control system can be transformed to Figure 3-3 [38]. 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.

Table 3-I The motor specifications and the controller parameters

	Rated Voltage	100[V]
	Rated Current	0.66[A]
R_a	Rotor Resistance	2.8[Ω]
L_a	Rotor Inductance	0.0011[H]
J_l	Motor Inertia	$4.96e^{-6}$[N·m·s^2]
D_l	Friction Coefficient	$3.543e^{-3}$[N·m·s]
K_T	Torque Const.	0.157 [N·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
$\triangle u_{in}$	Quantization (input)	10/256 [V]
$\triangle u_{out}$	Quantization (output)	0.5π/4096 [rad]
	Gear radio	0.02

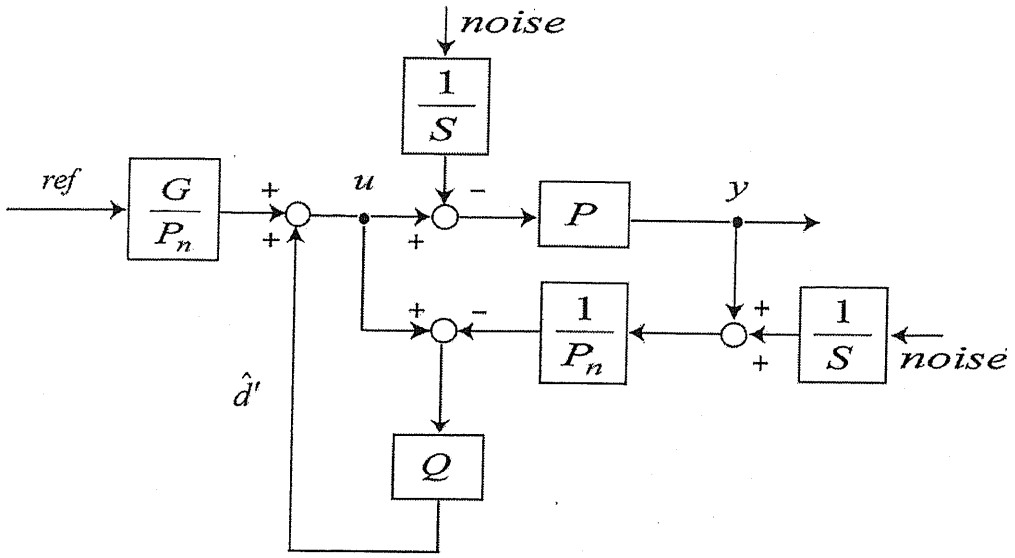


Figure 3-3: Disturbance observer type equivalent block diagram

3.2. Simulation

The simulation results for the noise over channel1 are shown in Figure 3-4. The simulation results for noise over channel1 and channel2 are shown in Figure 3-5. Figure 3-4 shows that the controller can compensate for noise over channel1 because controller C_2 has an integrator. However, as shown in Figure 3-5, the system does not compensate for the noise over channel2. Theoretically, the noise over channel2 is not observable. As a result, channel3 is added to compensate for the noise over channel2. The system configuration is shown in Figure 3-7. Because multi-bit format digital signals can be shaped into a one-bit format with sigma-delta modulation, the wiring costs are much lower than for some parallel communication schemes. Here, an n-bit parallel output signal from a servomotor side is transformed to a one-bit serial signal every n sampling periods, and the controller side receives a one-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 choose the best signal as the feedback signal from the servomotor. Figure 3-8 and Equation 3.9 show the compensation process for noise over channel3.

$$u(t) = \begin{cases} u_{channel2}(t) & if \quad t = nT \quad and \quad \left|u_{channel3}(t) - u_{channel2}(t)\right| > n \\ u_{channel3}(t) & if \quad t = nT \quad and \quad \left|u_{channel3}(t) - u_{channel2}(t)\right| \le n \\ u_{channel2}(t) & if \quad t \neq nT \end{cases} \tag{3.9}$$

Here, the maximum variation every nth sampling period is assigned as the threshold. Therefore, a threshold, n is chosen, that is the same as the bit-width in channel3. Figure 3-6 shows the results when noise affects all channels. Clearly, the system can compensate for noise over all channels.

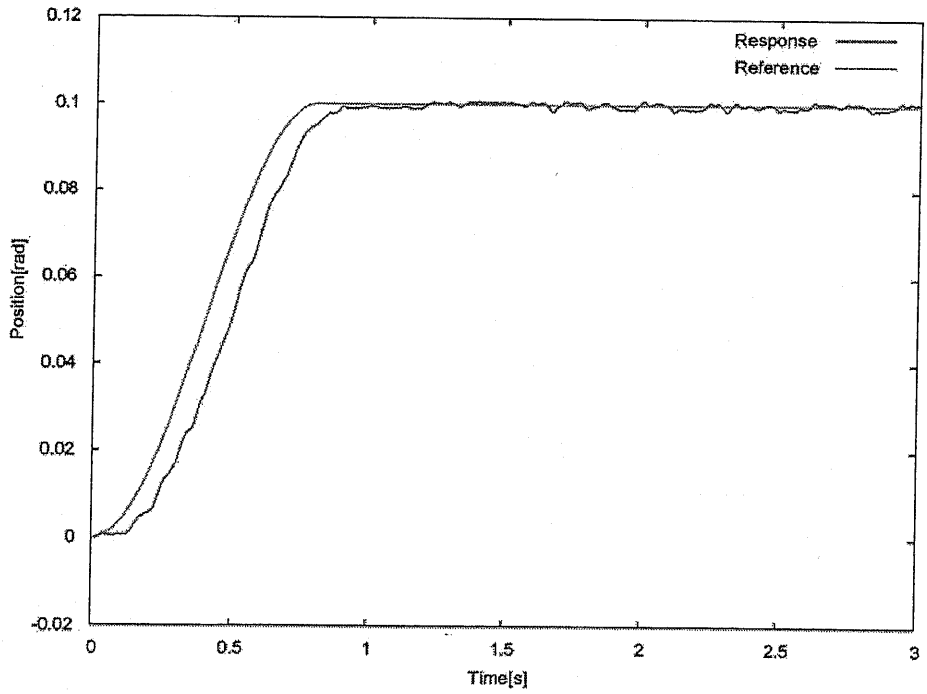


Figure 3-4: Response of the system with noise over channel1.

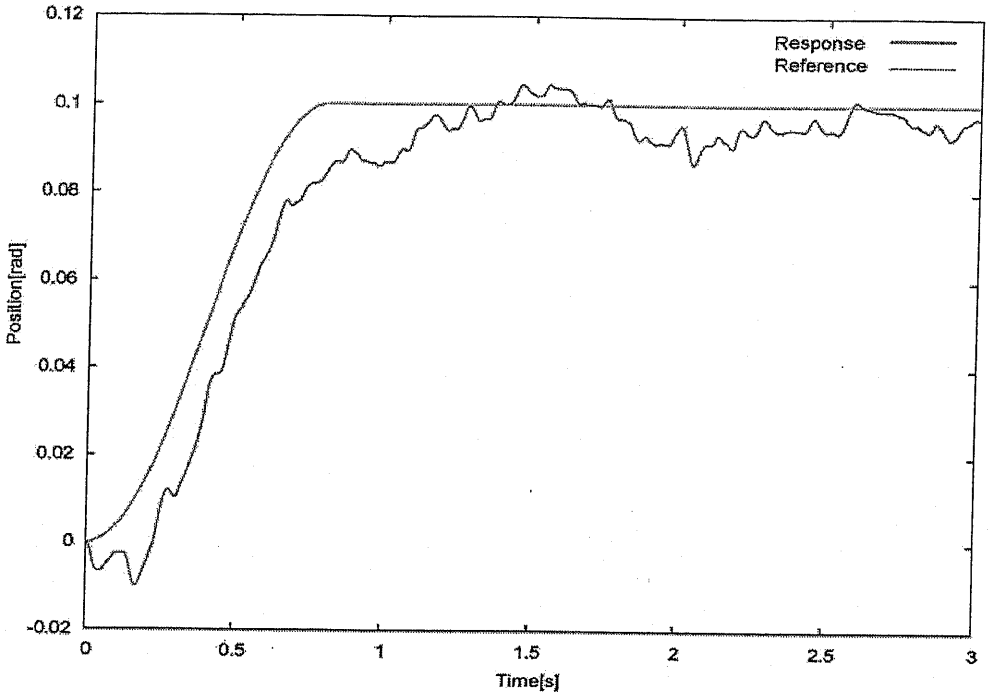


Figure 3-5: System response with noise over two transmission channels.

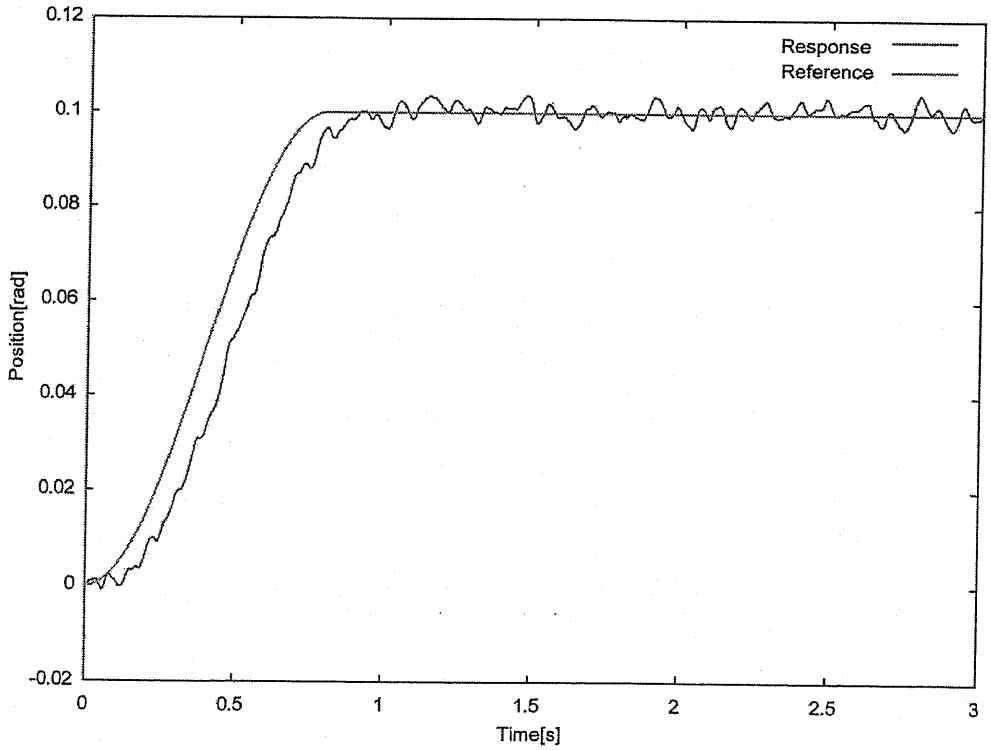


Figure 3-6: System response using noise compensation over three channels.

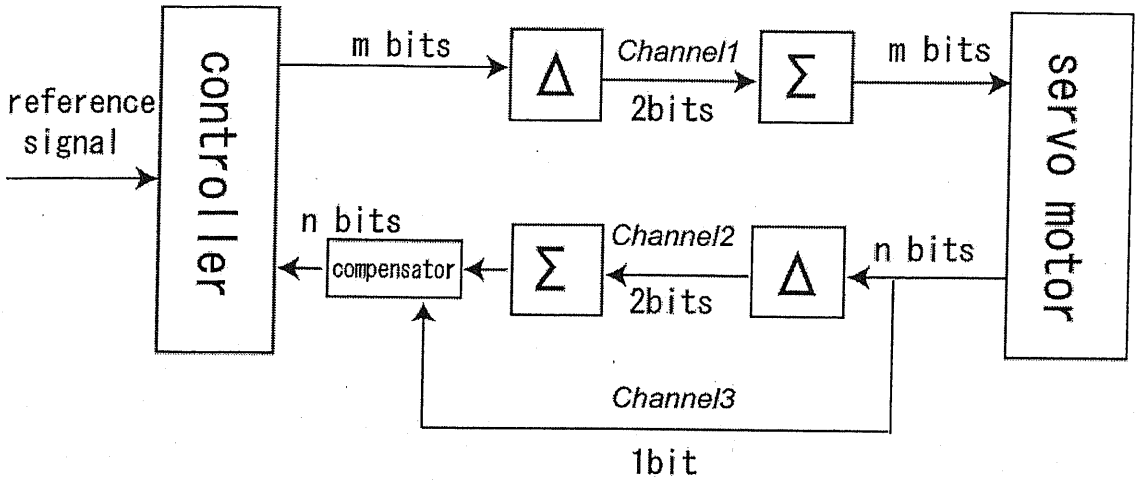


Figure 3-7: Noise-compensation model of channel2.

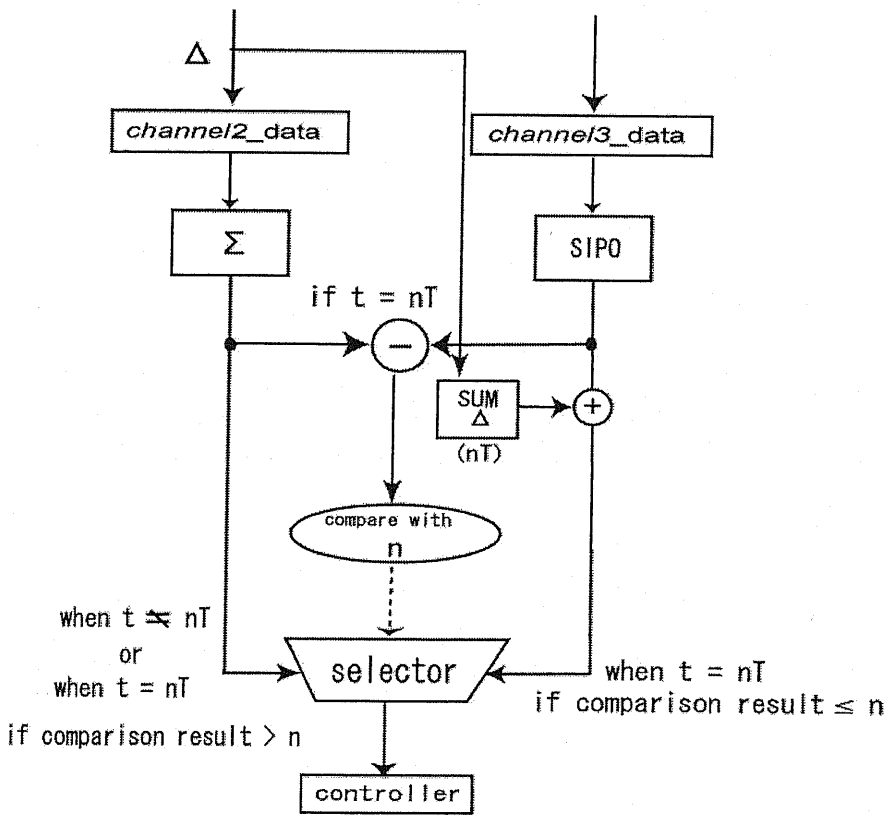


Figure 3-8: Noise-compensation method in feedback channel.

Chapter 4

4 Application to Servo Motor Control with Considering the Current Control Loop

In chapter three, sigma-delta modulation is applied on an AC servo control system. However, the current control loop of the system is employed by a servo amplifier, which limits the performance of the proposed high-speed control systems. Hence, the aim of this chapter is to develop a high-speed control and communication system while considering the current control loop for the AC servo machine, which is driven by a 100 kHz PWM inverter.

4.1. System configuration

Figure 4-1 shows the block diagram of the development of the control and communication system with communication channels connecting the controller and the controlled plant. The controlled plant is represented by an AC servomotor. The control system is divided into two parts: one is the position control system that is located at the remote side, and the other is the current control system, which is located on the local side. The communication system transmits the current reference and position information using sigma-delta modulation. 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 position controller.

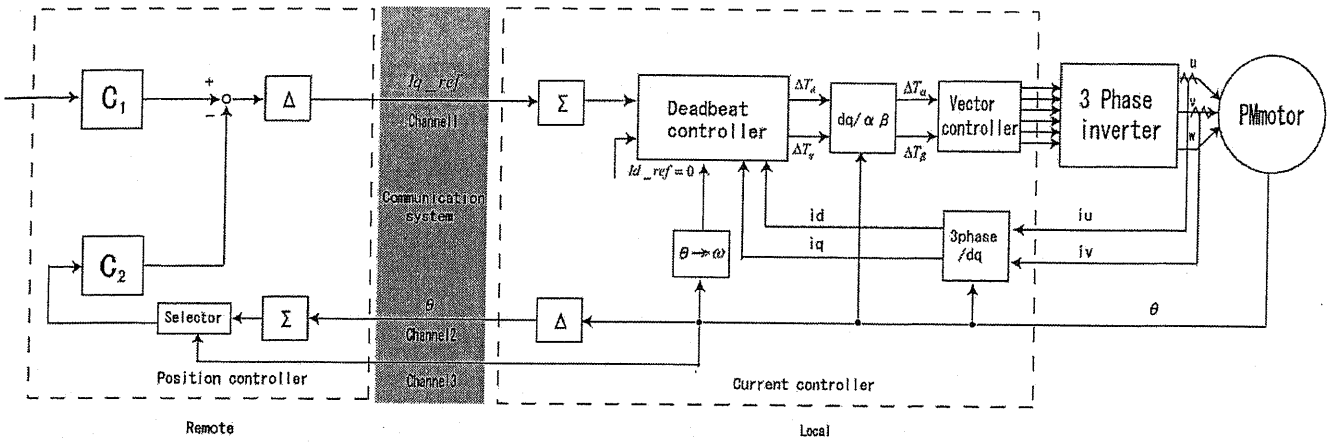


Figure 4-1: The block diagram of the proposed system.

4.2. Current controller design

4.2.1. Deadbeat controller

The current control loop is the most fundamental and important control loop for any AC drive system. If the current control loop has a very fast and accurate transient response, then, with appropriate reference currents, the machine can be regarded as an ideal torque source, as far as the outer loops are concerned. In this paper, the deadbeat control method [31], [54] is utilized to control the firing time of the switches of the PWM inverter, and the space vector control method is utilized to fire the switches. The PM motor is considered to be the controlled plant, which is driven by the PWM inverter. The d-q axis model of the controlled plant is expressed as

$$\begin{bmatrix} v_{da} \\ v_{qa} \end{bmatrix} = \begin{bmatrix} R_a + pL_a & -\omega_{re}L_a \\ \omega_{re}L_a & R_a + pL_a \end{bmatrix} \begin{bmatrix} i_{da} \\ i_{qa} \end{bmatrix} + \begin{bmatrix} 0 \\ \omega_{re}\Phi_{fa} \end{bmatrix} \tag{4.1}$$

where R_a is the rotor resistance, L_a is the rotor inductance, $p = d/dt$ is the difference figure,

ω_{re} is the angular velocity of the magnetic, and Φ_{fa} is the CEMF constant. For the PM-motor model d-axis current $i_{da}=0$ is kept by the current controller and the reluctance $L_d=L_q$ is given, so the reluctance torque is nonoccurrence. The model of q-axis can be expressed as DC motor. The function is

$$P_n=\frac{K_T}{(Js+B)s} \tag{4.2}$$

and the state function is

$$\dot{x}=Ax+Bu \tag{4.3}$$

The discretization model based on the switch firing time of the PWM inverter is

$$x(k+1)=Fx(k)+G\Delta T(k) \tag{4.4}$$

Where F and G are

$$F=\begin{bmatrix}F_{11} & F_{12} & F_{13}\\ F_{21} & F_{22} & F_{23}\\ F_{31} & F_{32} & F_{33}\end{bmatrix}=e^{AT}$$
$$G=\begin{bmatrix}G_1\\ G_2\\ G_3\end{bmatrix}=e^{AT/2}BE \tag{4.5}$$

The state variables are

$$\begin{aligned}x_1&=i_q(k)\\ x_2&=\theta_{re}(k)\\ x_3&=\omega_{re}(k)\end{aligned} \tag{4.6}$$

E is the input DC voltage of the PWM inverter. From Equation 4.3, we can obtain the q-axis current function

$$i_q(k+1)=F_{11}i_q(k)+F_{12}\theta_{re}(k)+F_{13}\omega_{re}(k)+G_1\Delta T_{iq}(k) \tag{4.7}$$

where $i_q(k+1)$ is the current reference of q-axis. Then the firing time of the switch is

$$\Delta T_q = \frac{i_{qref}(k+1) - \left(F_{11} i_q(k) + F_{12}\theta_{re}(k) + F_{13}\omega_{re}(k)\right)}{G_1} \tag{4.8}$$

With the same method, we can obtain the d-axis firing time of the switches from equation 4.1.

4.2.2. Space vector modulation

Once $\Delta T_q(k)$ and $\Delta T_d(k)$ have been evaluated using the deadbeat control method, they are converted into $\Delta T_\alpha(k)$ and $\Delta T_\beta(k)$ by Equation 4.9.

$$\begin{bmatrix}\Delta T_\alpha(k)\\ \Delta T_\beta(k)\end{bmatrix} = \begin{bmatrix}\cos(\theta(k)) & -\sin(\theta(k))\\ \sin(\theta(k)) & \cos(\theta(k))\end{bmatrix}\begin{bmatrix}\Delta T_d(k)\\ \Delta T_q(k)\end{bmatrix} \tag{4.9}$$

Then, $\Delta T_\alpha(k)$ and $\Delta T_\beta(k)$ are converted into the switching times for the three phases of the inverter. Using the space vector modulation method, the inverter is to be fired with a double edge modulation firing pattern.

Figure 4-2 shows the convention method [54] used here for the firing sequence of a three phase inverter. The notation of the space vectors, such as (1, 0, 0) for V_1, indicates which switch is closed on the inverter. The firing time of each switch can be calculated from simple trigonometry, as shown in Figure 4-2. The conditions required for particular sectors, the firing order of the phases and the times for the phase firings appear in Table 4-I [33].

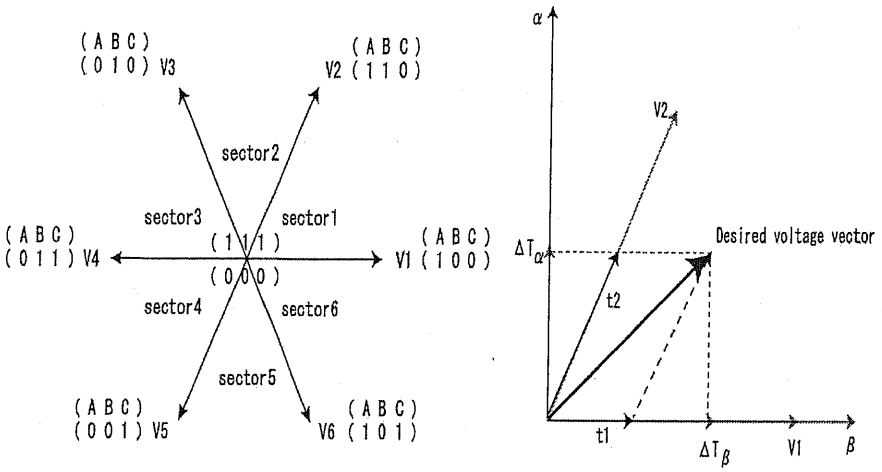


Figure 4-2: Space vector switching sequence for PWM inverter

Table 4-I 3 Phase Inverter Firing Times For Various Sectors

	Condition for sector	Firing Order	t_1	t_2
Sector1	$\Delta T_\alpha > 0; \Delta T_\beta > 0; \Delta T_\beta > \sqrt{3}\Delta T_\alpha$	$V_0V_1V_2V_1V_0$	$\Delta T_\alpha - \Delta T_\beta/\sqrt{3}$	$2\Delta T_\beta/\sqrt{3}$
Sector2	$\Delta T_\beta > 0; \Delta T_\beta \geq \sqrt{3}\lvert\Delta T_\alpha\rvert$	$V_0V_3V_2V_3V_0$	$-\Delta T_\alpha + \Delta T_\beta/\sqrt{3}$	$\Delta T_\alpha + \Delta T_\beta/\sqrt{3}$
Sector3	$\Delta T_\alpha > 0; \Delta T_\beta \geq 0; \Delta T_\beta < \sqrt{3}\lvert\Delta T_\alpha\rvert$	$V_0V_3V_4V_3V_0$	$2\Delta T_\beta/\sqrt{3}$	$-\Delta T_\alpha - \Delta T_\beta/\sqrt{3}$
Sector4	$\Delta T_\alpha < 0; \Delta T_\beta < 0; \Delta T_\beta > \sqrt{3}\Delta T_\alpha$	$V_0V_5V_4V_5V_0$	$-2\Delta T_\beta/\sqrt{3}$	$-\Delta T_\alpha + \Delta T_\beta/\sqrt{3}$
Sector5	$\Delta T_\beta < 0; \lvert\Delta T_\beta\rvert \geq \sqrt{3}\lvert\Delta T_\alpha\rvert$	$V_0V_5V_6V_5V_0$	$-\Delta T_\alpha - \Delta T_\beta/\sqrt{3}$	$\Delta T_\alpha - \Delta T_\beta/\sqrt{3}$
Sector6	$\Delta T_\alpha > 0; \Delta T_\beta < 0; \lvert\Delta T_\beta\rvert < \sqrt{3}\Delta T_\alpha$	$V_0V_1V_6V_1V_0$	$\Delta T_\alpha + \Delta T_\beta/\sqrt{3}$	$-2\Delta T_\beta/\sqrt{3}$

4.3. Position controller design

In the position control system, the type 1 servo controller can compensate for step disturbances on the input, which is introduced in chapter 3. The current to position transfer function of controlled plant is expressed as

$$P_n = \frac{K_T}{(Js + B)s} \tag{4.10}$$

where K_T is the torque constant, B is the friction coefficient and J is the motor inertia. Using the same control law in chapter 3, the two degrees of freedom controller is

$$C_1 = \frac{G}{P_n(1-Q)} = \frac{(B+Js)(s^3 + 2s^2\omega_c + 2s{\omega_c}^2 + {\omega_c}^3)}{K_T s(s+2\omega_c)(1+s^2\tau^2 + 2s\tau\xi)} \tag{4.11}$$

$$C_2 = \frac{Q}{P_n(1-Q)} = -\frac{(B+Js)(2s+\omega_c){\omega_c}^2}{K_T s(s+2\omega_c)} \tag{4.12}$$

where τ is the response-time constant and ω_c is the damped frequency, G, Q are the command-input response and the complementary sensitivity function respectively, whose setup was reported in chapter 3.

4.4. Simulation

The input channel quantization value, Δu_{in}, and the output channel quantization value, Δu_{out}, are set to 8.0/256 [A] and π/4096 [rad] respectively. The result with the step position reference of 1 [rad] is shown in Figure 4-3. From the response of the motor, we can confirm a reliable system response.

The simulation result for the noise over channel1 is shown in Figure 4-4. This result shows the designed position controller can compensate for the noise over channel1 because the controller C_2 can estimate the equivalent input disturbance caused by the noise. However, the controller does not compensate for the noise over channel2, as shown in Figure 4-5, because the noise over channel2 is unobservable. Thus, the channel3 is added to compensate for the noise over channel2. This compensation scheme was reported in chapter three. Figure 4-6 shows the result when noise affects all channels. Clearly, the system can compensate for noise over all channels and can provide the desired position response.

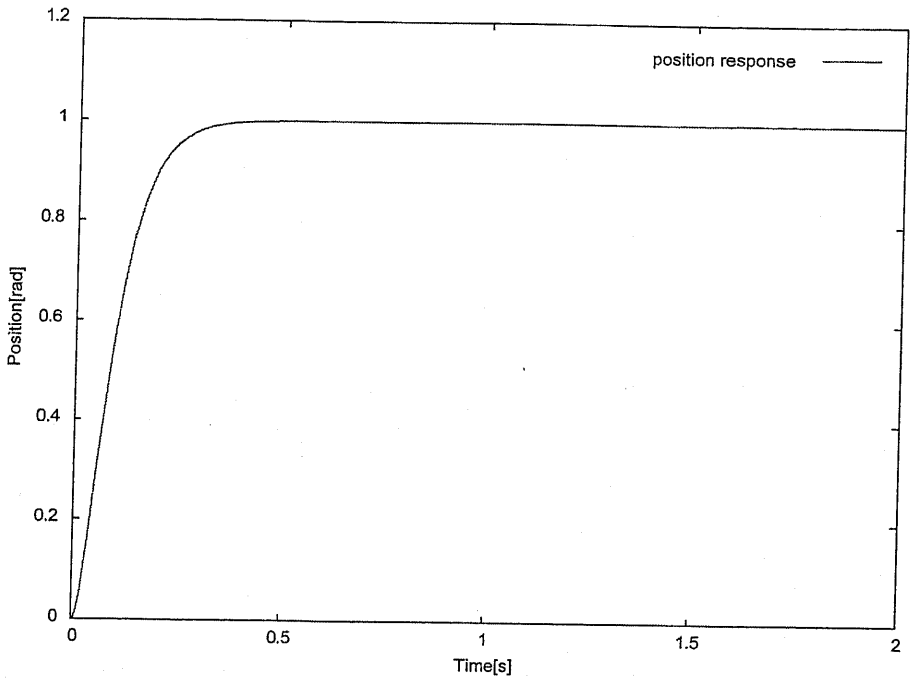


Figure 4-3: Simulation result for position response

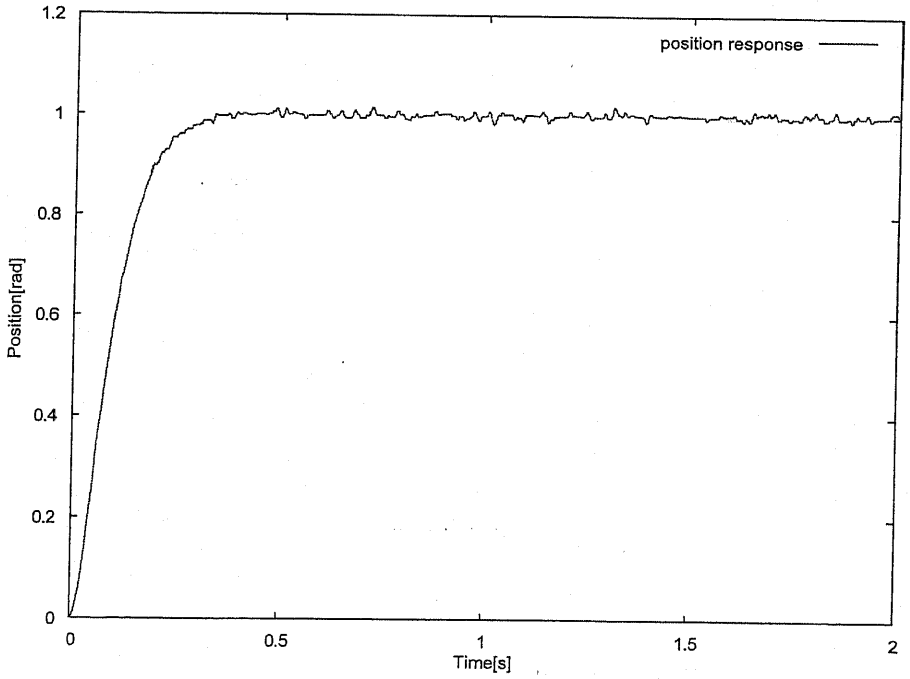


Figure 4-4: Position response with noise over channel1

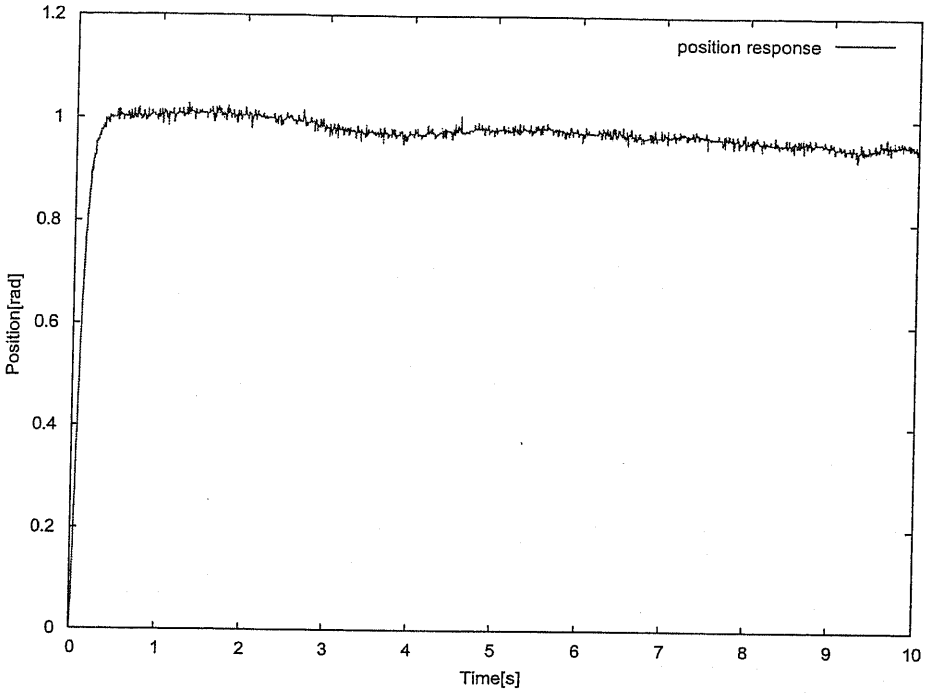


Figure 4-5: Position response with noise over channel2

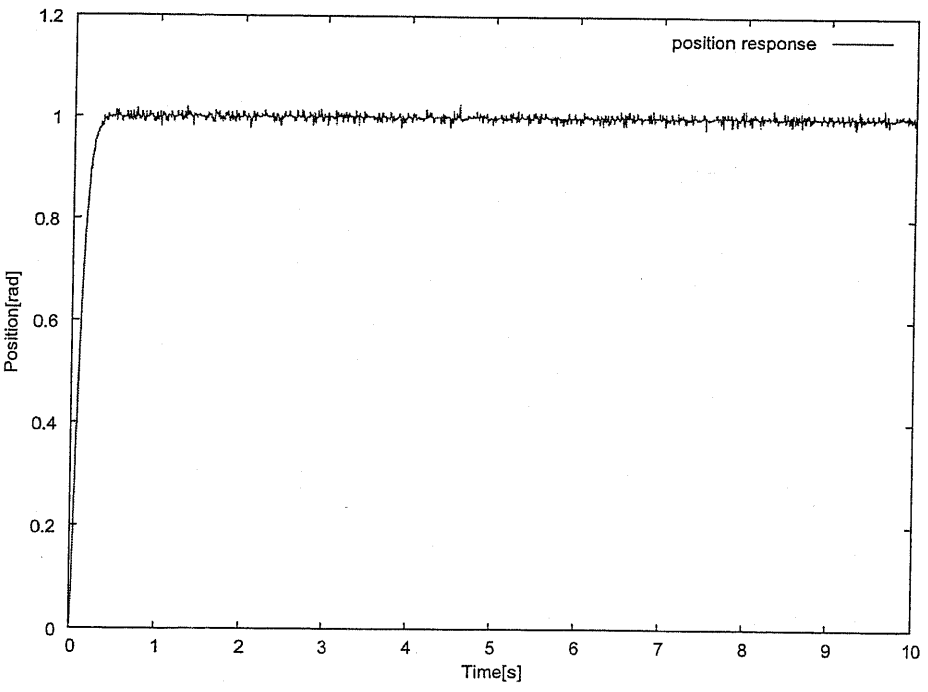


Figure 4-6: Position response with noise over all channels

Chapter 5

5 Bit Error Rate Analysis for proposed communication system

For a control system with quantized feedback control values being communicated from a controller to the plant over a noisy feedback channel, randomly occurring bit errors will degrade the system's performance. The extent of this degradation will depend on a number of factors, including how noisy the channel is and whether control levels have been coded in such a way that no single bit error in the specification of a control value will cause the plant to use a value that is significantly different from the one specified. Table 5-I [51] is a list of noise generating devices common to manufacturing environments. In a motor system, the largest source of noise is electrical fast transient (EFT), which is line disturbances caused by switching transients from nearby relays. A list of different physical transmission media is given in Table 5-II [52]. According to the standard IEC 61508, the residual error requirements are given in Table 5-III [8]. Here, SIL3 is the highest level found in traditional manufacturing and process applications, which means that the average probability of failure on demand per hour. For the proposed system, all three channels transmit 5 bit of data per cycle. In a single second, 500 kbit is transmitted, and in an hour, 1.8 Gbit is transmitted over the communication system. To satisfy the SIL3, the error probability of transmitting one bit will be less than

0.56×10^{-16}.

Table 5-I Sources of Industrial Noises

Type	Noise	Coupling mechanism
Motors	EFT	Local Ground, conducted
Relays and Contactors	EFT	Radiated, Conducted
Welding	EFT, Induction	Radiated magnetic
Heating	EFT	Local Ground, Conducted, Radiated
Radio Communications	Radio Frequency	Radiated

Table 5-II Typical bit error rates for various medias

Physical media	Error probability
Cellular radio	10^{-2} to 10^{-3}
Infrared	10^{-4} to 10^{-6}
Copper double wire	10^{-4} to 10^{-6}
Twisted pair (Differential)	$\leq 10^{-7}$
Fiber Optics	$\leq 10^{-9}$

Table 5-III Definition of safety integrity levels (SIL)

SIL3	10^{-8} to 10^{-7}
SIL2	10^{-7} to 10^{-6}
SIL1	10^{-6} to 10^{-5}

(The average probability of failure on demand per hour)

In this chapter, the error distribution is presented, occurring by the bit error over the digital

serial communication channels.

The definition of bit error rate (BER) is the ratio of the number of erroneous bits detected to the number of transmitted bits. This can be expressed mathematically as

$$p = \frac{\varepsilon}{n} \tag{5.1}$$

where the number of bit errors is ε, the total number of bits transmitted is n, and the bit error rate is set to the error probability of the twisted pair wire, $p = 10^{-7}$

5.1. Error distribution over sigma-delta channel

In our proposed communication system, the control values are compressed by the modified sigma-delta modulation and are transmitted by the digital serial communications. The output of delta modulation exists in three states, which are defined as 1, 0 and -1. As the single over the digital serial communication systems, a two-bit binary number for each packet is transformed. The two-bit binary number is coded by the following operation:

$$v_k = \begin{cases} 00 & if \quad \Delta = 0 \\ 01 & if \quad \Delta = 1 \\ 11 & if \quad \Delta = -1 \end{cases} \tag{5.2}$$

Note that 10 is not used in this coding process, but it occurs anyway due to the noise. In the event 10 occurs, the receiver will regard it as 0. In the control system, the three state coding sequence can be assumed independent and identically distributed. Thus, the occurring probability for each two-bit binary coding sequence is

$$P(00) = P(01) = P(11) \tag{5.3}$$

According to these factors, the occurring error values for one packet are -2△, -△, 0, △ and 2△. For each coding sequence, the probabilities of each error value are

$$v_k = 01 \rightarrow \begin{cases} 00 & -\Delta & \frac{1}{3}p(1-p) \\ 10 & -\Delta & \frac{1}{3}p^2 \\ 11 & -2\Delta & \frac{1}{3}p(1-p) \end{cases} \tag{5.4}$$

$$v_k = 00 \rightarrow \begin{cases} 10 & 0 & \frac{1}{3}p(1-p) \\ 01 & \Delta & \frac{1}{3}p(1-p) \\ 11 & -\Delta & \frac{1}{3}p^2 \end{cases} \tag{5.5}$$

$$v_k = 11 \rightarrow \begin{cases} 00 & \Delta & \frac{1}{3}p^2 \\ 10 & \Delta & \frac{1}{3}p(1-p) \\ 01 & 2\Delta & \frac{1}{3}p(1-p) \end{cases} \tag{5.6}$$

For the above formula, v_k is the signal, which will be transmitted. The first column expresses the signal by occurring noise, the second column expresses the error value, and the third column shows the probability for each error value. Combining these, the error value distribution for one packet is

$$e_1 = \begin{cases} \frac{1}{3}p(1-p) & error = 2\Delta \\ \frac{1}{3}p(2-p) & error = \Delta \\ \frac{1}{3}p(1-p) & error = 0 \\ \frac{1}{3}p(1+p) & error = -\Delta \\ \frac{1}{3}p(1-p) & error = -2\Delta \end{cases} \tag{5.7}$$

Assuming that there are, at first, eight packets of data to be transmitted over the channel, if and only if one packet bit error occurs, the error value distribution is

$$P_1 = e_1(1-p)^{14}C_8^1 \tag{5.8}$$

where C_8^1 is defined as

$$\frac{8!}{1!(8-1)!} \tag{5.9}$$

Thus, if and only if n packets bit error occurs, the error value distribution is

$$P_n = e_n(1-p)^{16-2n}C_8^n \tag{5.10}$$

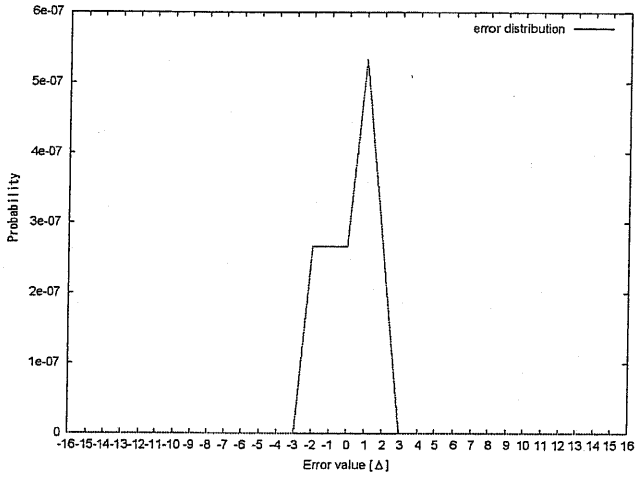
where e_n is the accumulation of e_1, and the cumulative process occurs 5^n times. The error value distributions for 1 to 8 packets occurring bit error are shown in Figure 5-1. In combination with the entire events occurring in the eight packets of data, the error distribution is shown in

Figure 5-2 and Table 5-IV, where the equation is

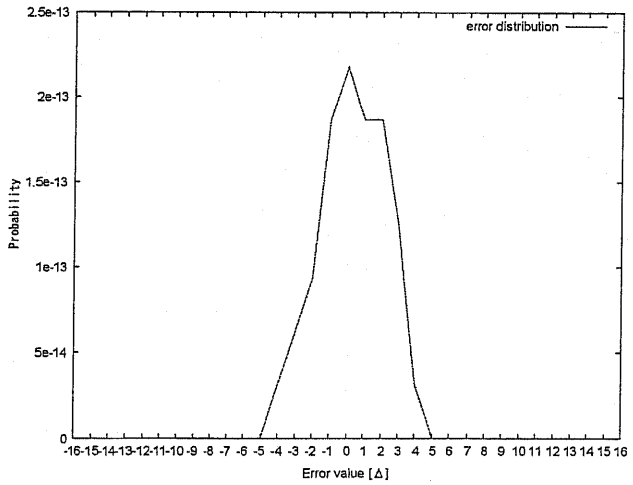
$$\sum_{n=1}^{8} C_8^n e_n (1-p)^{(16-2n)} + (1-p)^{16} = 1 \tag{5.11}$$

The probability for the error value being zero is

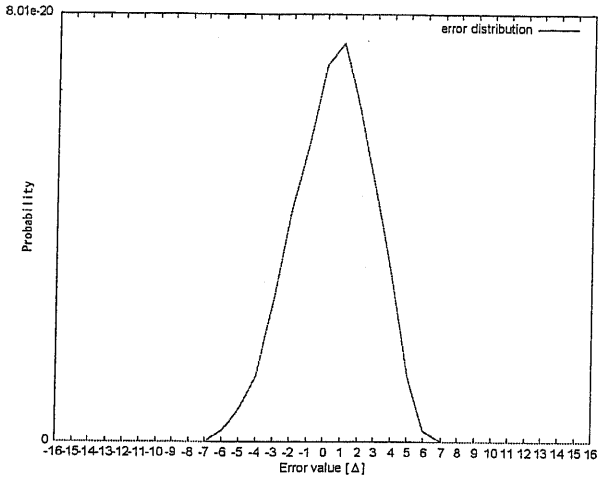
$$\sum_{n=1}^{8} C_8^n e_n (1-p)^{(16-2n)}\Big|_{error=0} + (1-p)^{16} = 0.9999987 \tag{5.12}$$



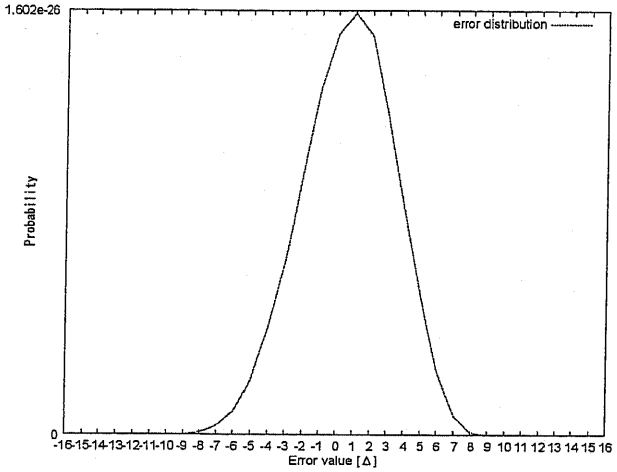
(a) Error distribution for 1 packet error occurring



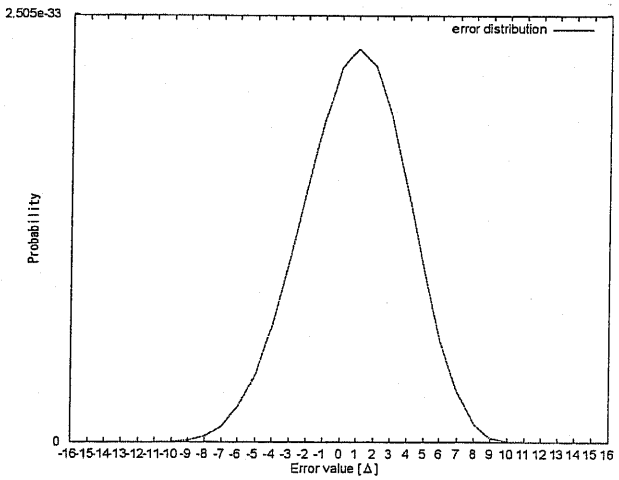
(b) Error distribution for 2 packets error occurring



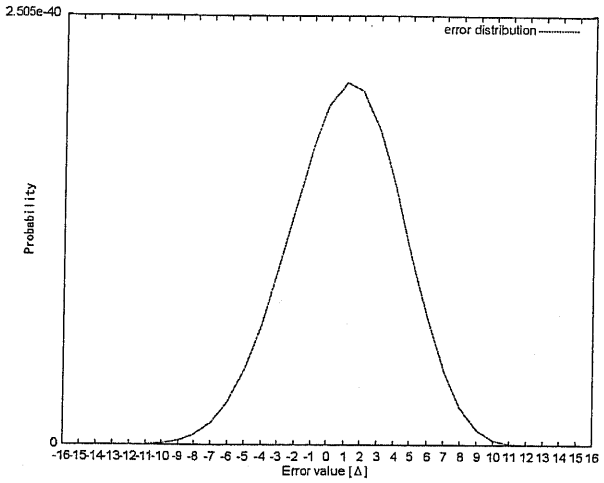
(c) Error distribution for 3 packets error occurring



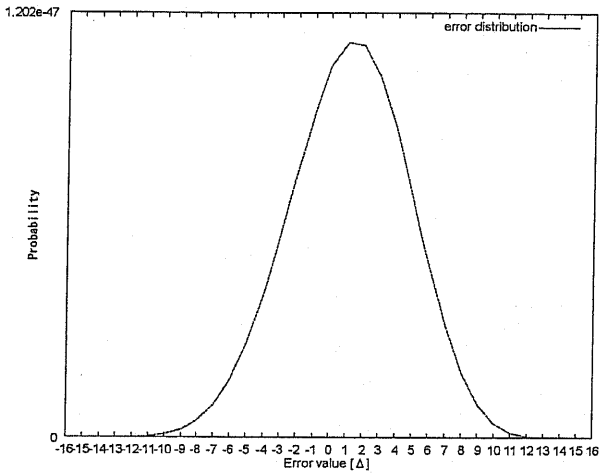
(d) Error distribution for 4 packets error occurring



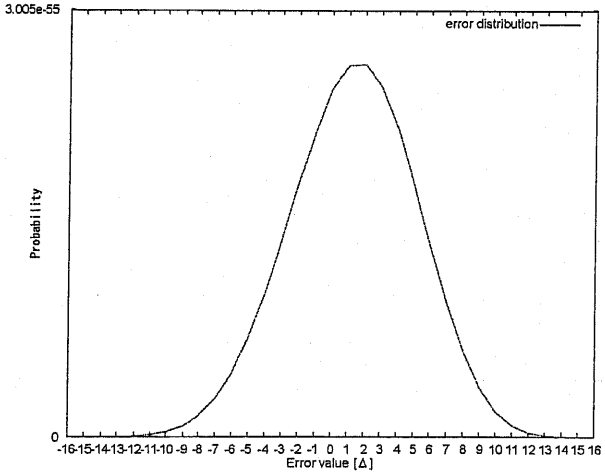
(e) Error distribution for 5 packets error occurring



(f) Error distribution for 6 packets error occurring



(g) Error distribution for 7 packets error occurring



(h) Error distribution for 8 packets error occurring

Figure 5-1: Error distribution for the error occurring 1 to 8 packets

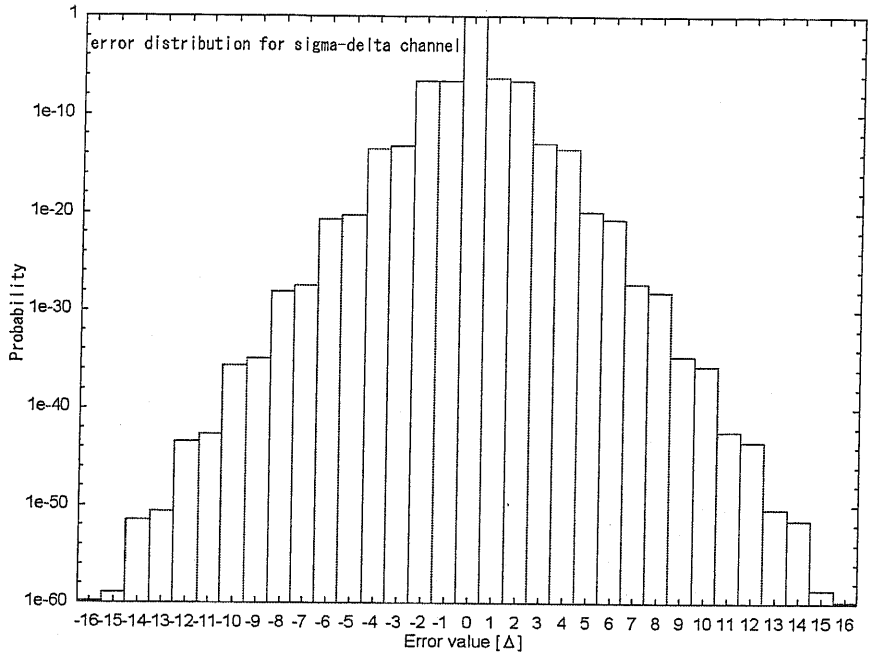


Figure 5-2: Error distribution for sigma-delta channel with the coding scheme (00, 01, 11).

Table 5-IV Error distribution for sigma-delta channel with coding 00, 01, 11($p = 10^{-7}$)

Error Value	Probability	Error Value	Probability
-16△	1.524e-60	16△	1.524e-60
-15△	1.219e-59	15△	2.438e-59
-14△	3.657e-52	14△	3.657e-52
-13△	2.560e-51	13△	5.121e-51
-12△	3.840e-44	12△	3.840e-44
-11△	2.304e-43	11△	4.609e-43
-10△	2.304e-36	10△	2.304e-36
-9△	1.152e-35	9△	2.304e-35
-8△	3.456e-29	8△	8.641e-29
-7△	3.456e-28	7△	6.913e-28
-6△	2.074e-21	6△	2.074e-21
-5△	6.222e-21	5△	1.244e-20
-4△	3.111e-14	4△	3.111e-14
-3△	6.222e-14	3△	1.244e-13
-2△	2.666e-7	2△	2.666e-7
-1△	2.666e-7	1△	5.333e-7
0	0.9999987		

Here, another two-bit binary number coding scheme is given as the flowing operation:

$$v_k' = \begin{cases} 00 & if \quad \Delta = 0 \\ 01 & if \quad \Delta = 1 \\ 10 & if \quad \Delta = -1 \end{cases} \tag{5.13}$$

Note that 11 is not used in this coding process. In this coding scheme, the receiver will regard it as 0. The other factors are identical to the above coding scheme, and the error value distribution for one packet is

$$e_1' = \begin{cases} \frac{1}{3}p^2 & error = 2\Delta \\ p(1-p) & error = \Delta \\ \frac{1}{3}p^2 & error = 0 \\ p(1-p) & error = -\Delta \\ \frac{1}{3}p^2 & error = -2\Delta \end{cases} \tag{5.14}$$

The error distribution with this coding scheme is given in Figure 5-3 and Table 5-V. The probability for the error value of zero is 0.9999987. Compared with the above coding scheme of 00, 01, 11, this coding scheme can achieve a lower probability for almost every error value except for the error values of 1, 0 and -1; however, from Equation 5.14 and Equation 5.7, we can see that it cannot provide a better performance than the above coding scheme at the error value of zero. In other words, the above coding scheme can achieve a much lower probability of an occurring error. It should be noted that, in Table 5-IV and Table 5-V, the probability values are the same for the error value of zero because the bit error rate is too small. For the bit error rate of 0.001, the probability with the coding schemes from Equation 5.2 and Equation 5.13, and the probability at the zero error value are 0.986765 and 0.9841773 respectively.

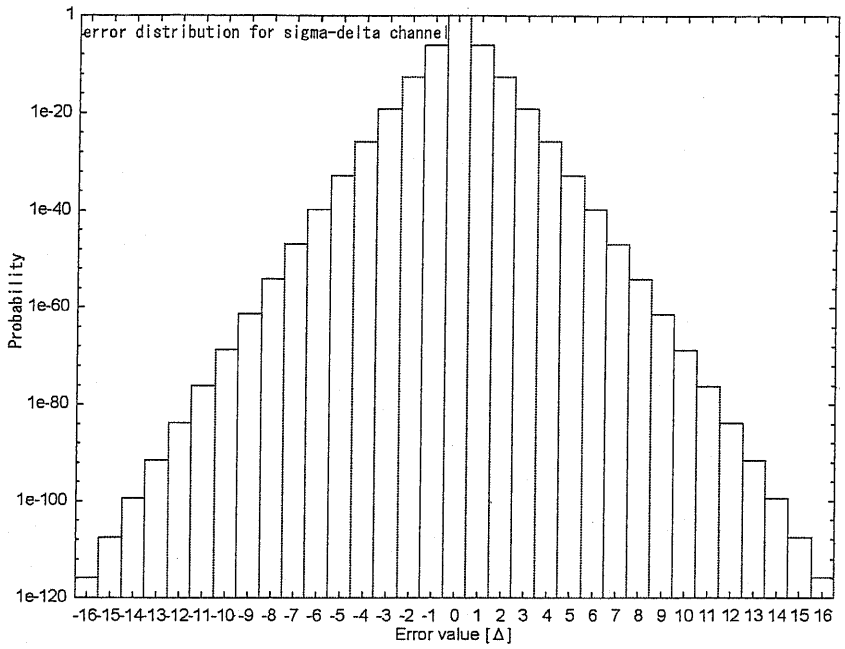


Figure 5-3: Error distribution for sigma-delta channel with the coding scheme (00, 01, 10).

Table 5-V Error distribution for sigma-delta channel with coding 00, 01, 10($p = 10^{-7}$)

Error Value	Probability	Error Value	Probability
-16△	1.524e-60	16△	1.524e-60
-15△	1.828e-59	15△	1.828e-59
-14△	3.657e-52	14△	3.657e-52
-13△	3.840e-51	13△	3.840e-51
-12△	3.840e-44	12△	3.840e-44
-11△	3.456e-43	11△	3.456e-43
-10△	2.304e-36	10△	2.304e-36
-9△	1.728e-35	9△	1.728e-35
-8△	8.641e-29	8△	8.641e-29
-7△	5.185e-28	7△	5.185e-28
-6△	2.074e-21	6△	2.074e-21
-5△	9.333e-21	5△	9.333e-21
-4△	3.111e-14	4△	3.111e-14
-3△	9.333e-14	3△	9.333e-14
-2△	2.666e-07	2△	2.666e-07
-1△	3.999e-07	1△	3.999e-07
0	0.9999987		

5.2. Error distribution over compensation channel

In this paper, the compensation channel is proposed to compensate for the noise over the feedback sigma-delta channel, which is unobservable by the controller. In the compensation channel, an n-bit parallel signal is transformed into a serial signal to be transmitted with n packets and an n sampling period. The first eight packets error occurring process is considered in this analysis process. Thus, the occurring error values due to the bit error are from -255 to 255. However, the values -8 to 8 are useful in the analysis. The other values are neglected because of the proposed noise compensation scheme. Thus, the occurring probabilities of the error values -8 to 8 are obtained as follows

$$p_{compensation\_channel} = \begin{cases} error = 0 \rightarrow (1-p)^8 \\ error = \pm 1, \pm 2, \pm 4, \pm 8 \rightarrow \sum_{n=1}^{8} 2^{-n} p^n (1-p)^{(8-n)} \\ error = \pm 3 \rightarrow \sum_{n=2}^{8} 2^{(-n+1)} p^n (1-p)^{(8-n)} \\ error = \pm 5 \rightarrow \sum_{n=3}^{8} 3 * 2^{-n} p^n (1-p)^{(8-n)} + 2^{-2} p^2 (1-p)^6 \\ error = \pm 6 \rightarrow \sum_{n=2}^{8} 2^{(-n+1)} p^n (1-p)^{(8-n)} \\ error = \pm 7 \rightarrow \sum_{n=3}^{8} 3 * 2^{-n} p^n (1-p)^{(8-n)} + 2^{-2} p^2 (1-p)^6 \end{cases} \quad (5.15)$$

and the distribution is shown in Table 5-VI. In combination with the sigma-delta channel, the error value distribution is

$$P = p_{compensation\_channel} \times \sum_{error=-8}^{8} p_{compensation\_channel} + p_{\Delta\Sigma\_channel} \times \left(1 - \sum_{error=-8}^{8} p_{compensation\_channel} \right) \quad (5.16)$$

Figure 5-4 and Table 5-VII shows the distribution of every error value. The probability of an error value being zero is 0.999992. Compared with the error value distribution of the sigma-delta

channel, shown in Figure 5-5, the error probabilities with the combined compensation channel are higher at the error values of $\pm 4\Delta, \pm 7\Delta, \pm 8\Delta$. However, significant benefits will be displayed in the following communication process because the compensation channel depends on a cycle process within every n sampling period. In this case, $n = 8$.

For the case of all channels, $n = 8$, 40-bits data are transmitted over channel in 8 sampling periods. To satisfy the SIL3, the error probability will be less than $0.56 \times 10^{-16} \times 40 = 2.24 \times 10^{-15}$. The error probability is 3.918×10^{-35} for the error values except for $-8\Delta \sim 8\Delta$. So, if the fault is occurred for the system when the error values except for $-8\Delta \sim 8\Delta$. The system satisfies the requirements of SIL3.

Table 5-VI Error distribution for compensation channel ($p = 10^{-7}$)

Error Value	Probability	Error Value	Probability
-8△	4.999e-8	8△	4.999e-8
-7△	3.749e-22	7△	3.749e-22
-6△	2.499e-22	6△	2.499e-22
-5△	3.749e-22	5△	3.749e-22
-4△	4.999e-8	4△	4.999e-8
-3△	2.499e-22	3△	2.499e-22
-2△	4.999e-8	2△	4.999e-8
-1△	4.999e-8	1△	4.999e-8
0	0.9999992		

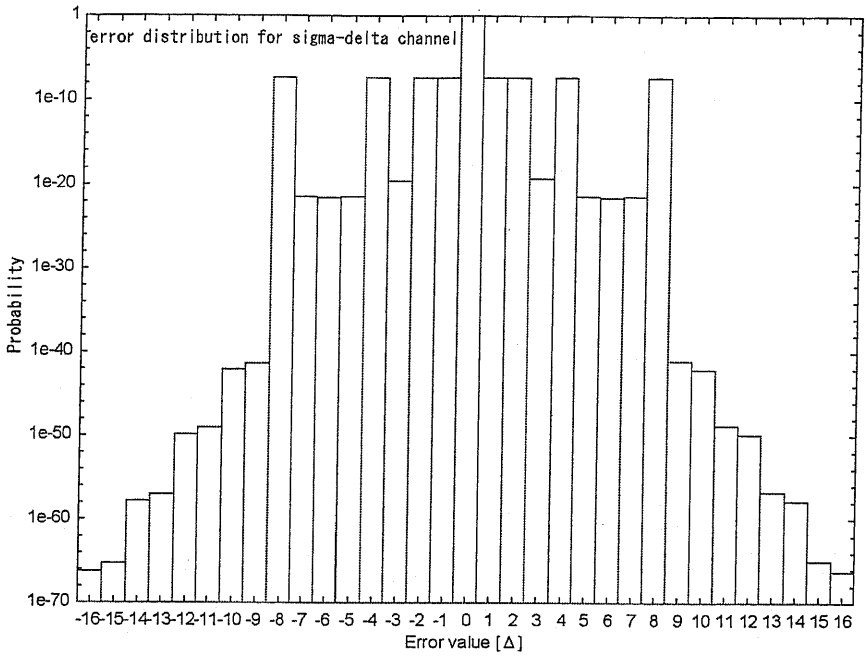


Figure 5-4: Error distribution for combination with compensation channel

Table 5-VII Error distribution with the Proposed Noise Compensation Method ($p = 10^{-7}$)

Error Value	Probability	Error Value	Probability
-16△	6.096e-67	16△	6.096e-67
-15△	4.877e-66	15△	9.754e-66
-14△	1.463e-58	14△	1.463e-58
-13△	1.024e-57	13△	2.048e-57
-12△	1.536e-50	12△	1.536e-50
-11△	9.218e-48	11△	1.843e-49
-10△	9.218e-43	10△	9.218e-43
-9△	4.609e-42	9△	9.218e-42
-8△	4.999e-8	8△	4.999e-8
-7△	3.749e-22	7△	3.749e-22
-6△	2.500e-22	6△	2.500e-22
-5△	3.750e-22	5△	3.750e-22
-4△	4.999e-8	4△	4.999e-8
-3△	2.513e-20	3△	5.002e-20
-2△	5.000e-8	2△	5.000e-8
-1△	5.000e-8	1△	5.000e-8
0	0.9999992		

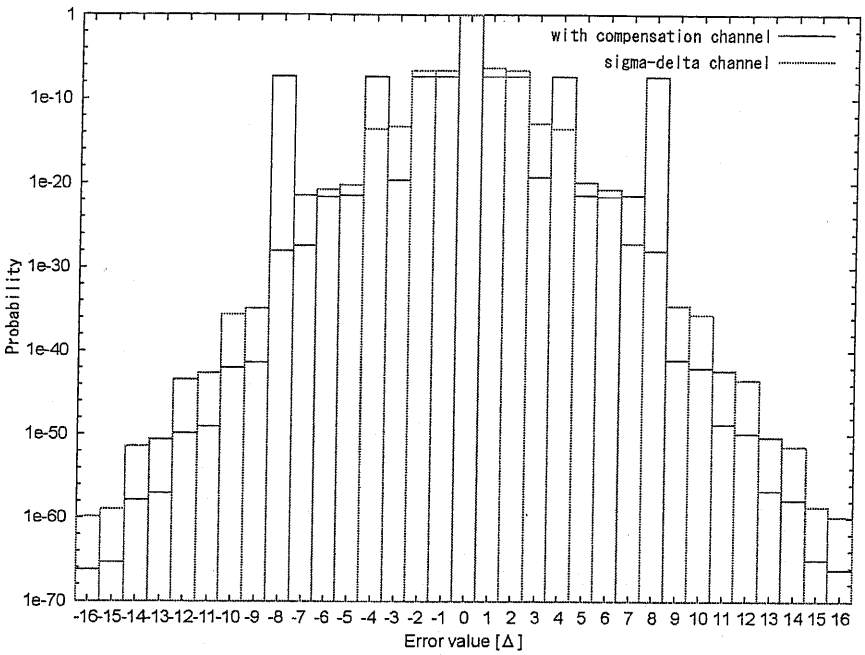


Figure 5-5: Error distribution for comparing with the sigma-delta channel

Chapter 6

6 FPGA Implementation

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. Speed performance of new components and flexibility inherent of all programmable solutions give today many opportunities in the field of digital implementation for industrial control systems [18]. FPGAs are programmable devices that allow us to implement the application directly in hardware and so we may exploit the parallelism inherent to many algorithms to decrease their processing time. DSPs and microcontrollers, on the other hand, are intrinsically sequential machines, which means that the tasks are necessarily executed sequentially which takes a longer processing time to accomplish the same task.

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 [18], [19], [20]. These benefits have resulted in the application of FPGAs in various fields such as telecommunication, signal processing, embedded control systems [25], robotics, electrical machine control systems [35], [37] and motion control [5], [30], [44]. This paper focus on networked control systems implemented with FPGAs—the entire control and communication system is implemented in FPGAs.

The control system is described in VHSIC hardware description language (VHDL). The basic structure of the designed circuits in the FPGAs is shown in Figure 6-1. On left side, the main

controller module is implemented in FPGA1, one part of the sigma-delta modulation module, and the communication interface circuits. On the right side, the motor counter module is implemented in FPGA2, the other part of the sigma-delta modulation, and the communication interface circuits. To implement this system in the experiment, we added channel4 to transform the output of the controller in the same way as channel3.

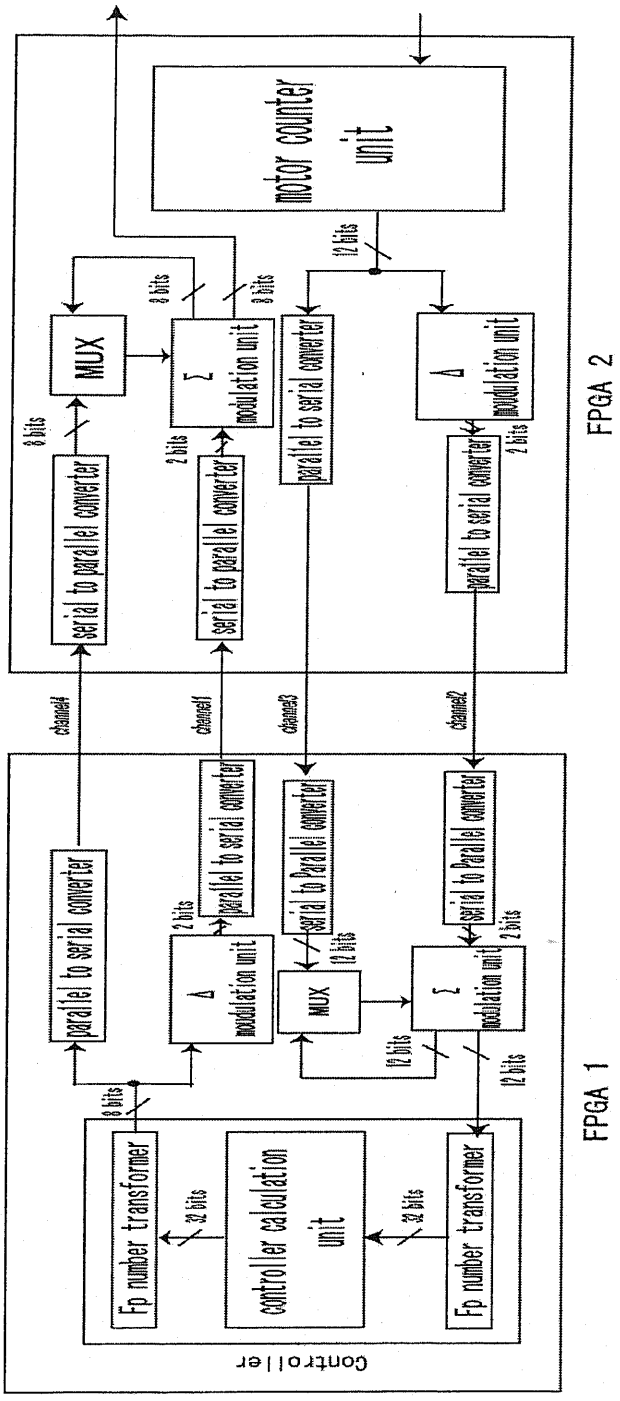


Figure 6-1: The basic configuration in FPGAs.

6.1. Communication system implementation

The communication system circuits in the FPGA are divided into three main parts: 1) the motor counter unit, 2) the sigma-delta modulation unit and 3) the communication interface unit

6.1.1. Motor-counter unit

The 4x multiplication counter circuit was designed 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 determining unit, as shown in Figure 6-2 where *cp* is the clock pulse. In the counter circuit, the comparator tests the changes of pulse of the *a* and *b* phases from the servomotor encoder in each clock pulse. The output of comparator is 0 or 1, which triggers the up/down counter. In addition, the motor rotation direction can be determined by the relation of the *a* and *b* phases, by the up/down determining unit [5].

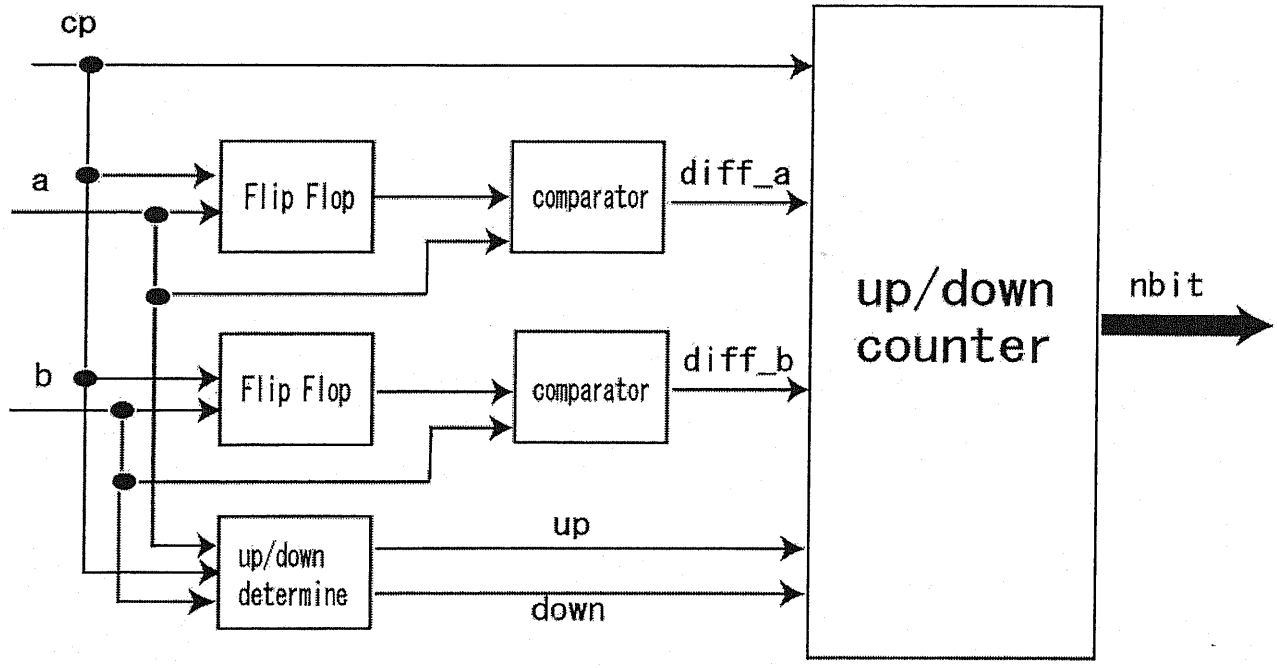


Figure 6-2: Motor-counter unit.

6.1.2. Sigma-delta modulation unit

Figure 6-3 shows the configuration of the delta modulator in FPGA, where *cp* is the clock pulse. Here, the comparator can test changes in the value of the input signal, which is latched by a flip-flop at each sample. The output of the comparator has three states, expressed as two bits—11, 00 and 01. The configuration of sigma modulation in the FPGA is shown in Figure 6-4, where *cp* is the clock pulse. The counter will be either up or down according to the output of the delta modulator, and the value is expressed in a *n*-bit signal as the output of sigma modulation

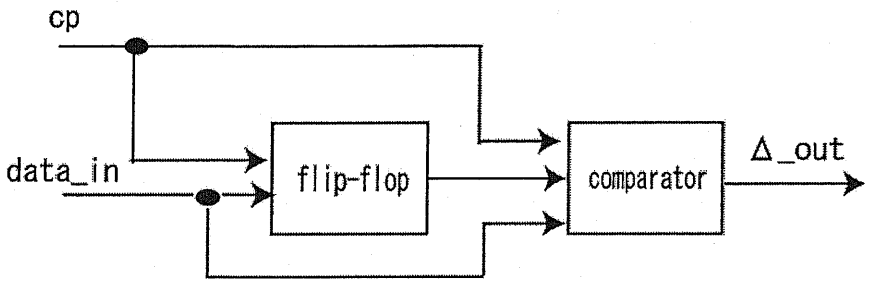


Figure 6-3: Configuration of delta modulation in FPGA.

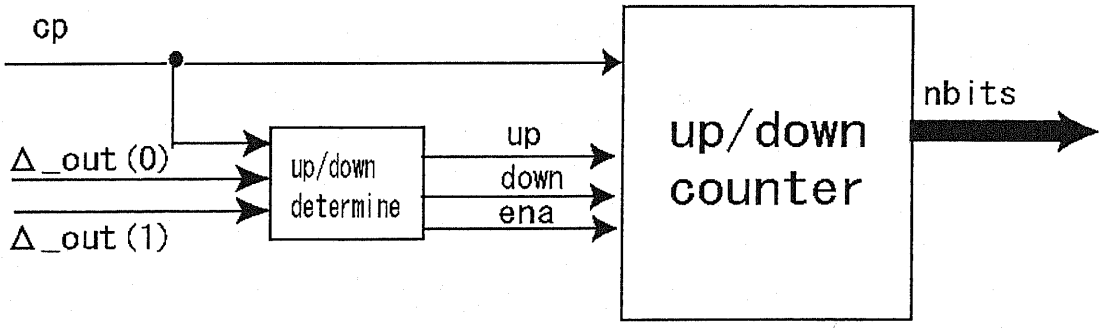


Figure 6-4: Configuration of sigma modulation in FPGA.

6.1.3. Communication interface

The unidirectional serial communication interface circuit was designed with FPGAs to reduce the cost of wiring. 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 request-to-send (RTS) 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 oversampling method is utilized to provide more detailed sampling periods than that of a single bit. The configuration of the transmitted data is shown in Figure 6-5, and the transmission process is shown in Figure 6-6. This communication method can synchronize both FPGAs to the same clock level, providing a high communication data rate over short distances. The circuits of receiver and driver are shown in Figure 6-7 and Figure 6-8.

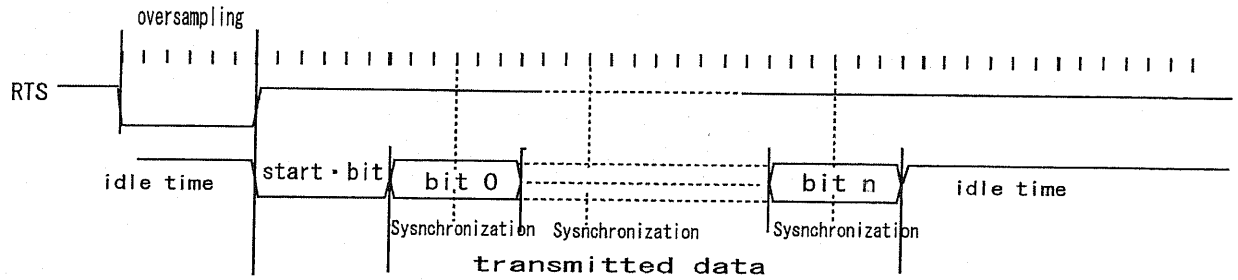


Figure 6-5: The configuration of transmitted data

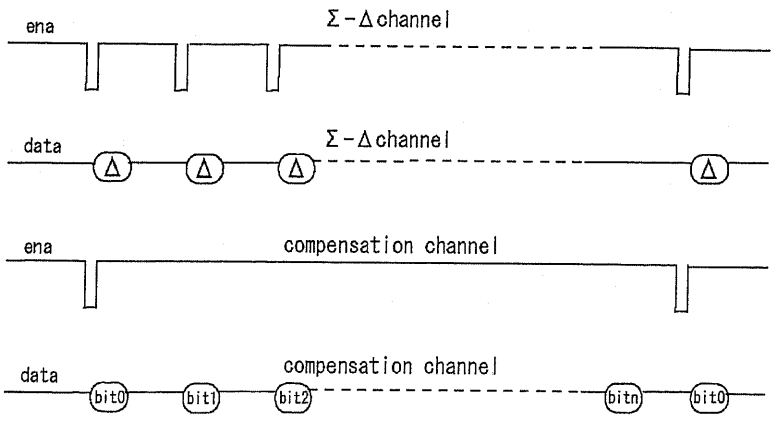


Figure 6-6: Data transmission process for sigma-delta channel and compensation channel.

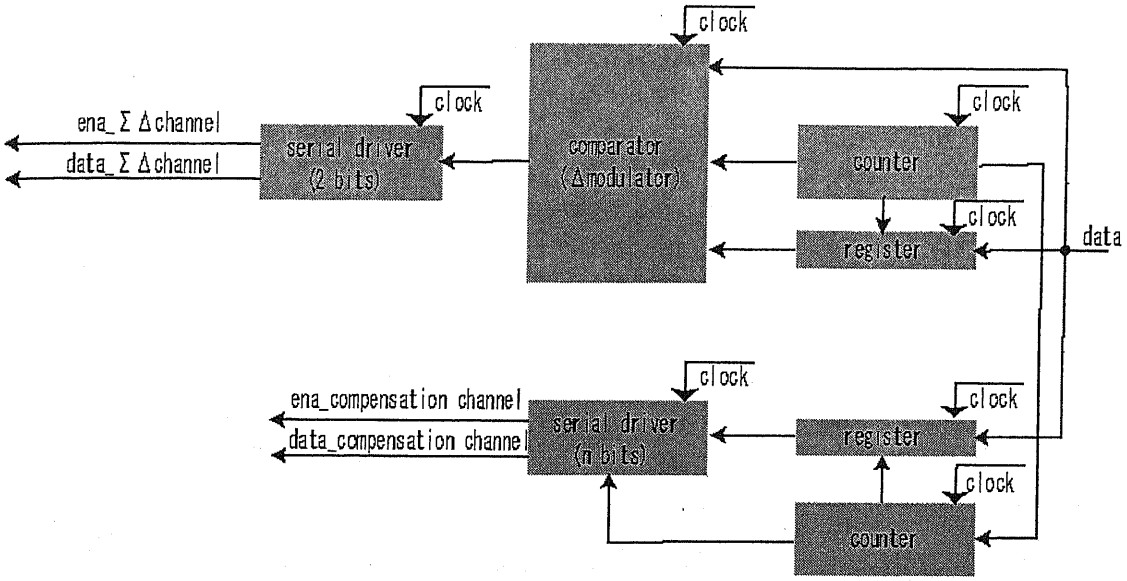


Figure 6-7: Driver circuit

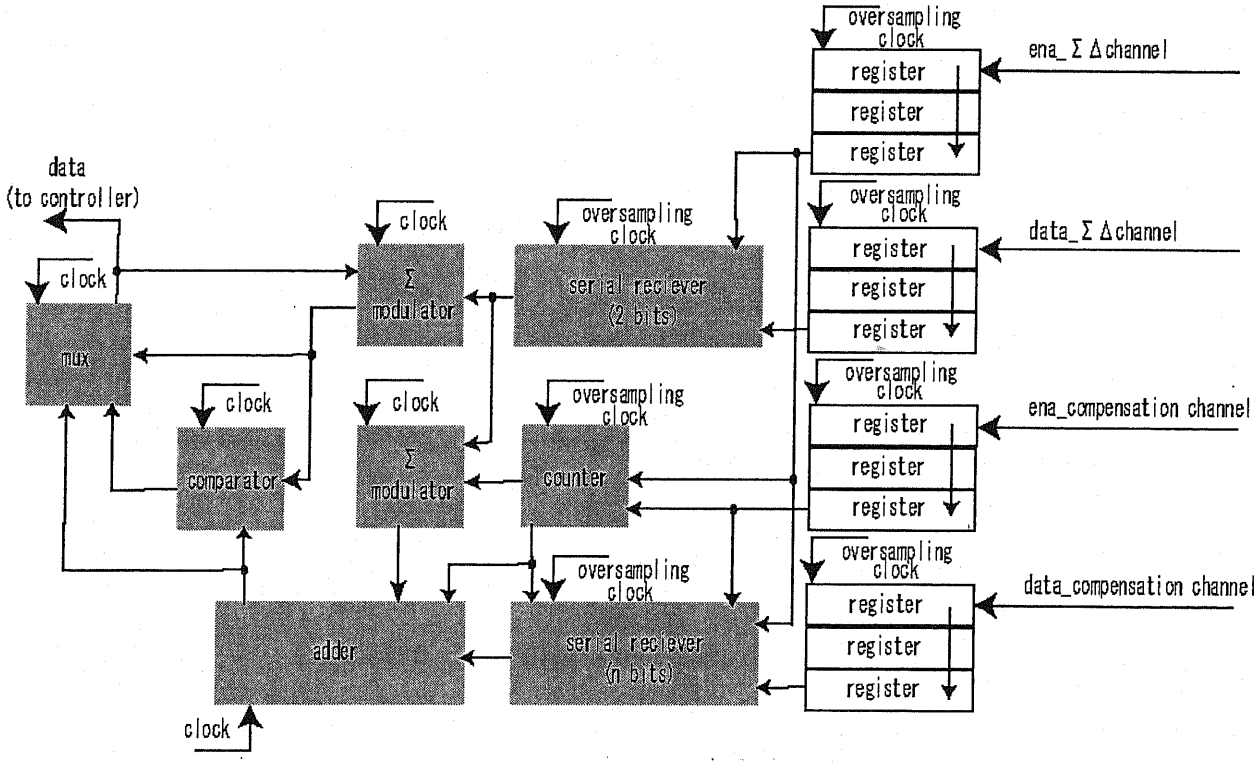


Figure 6-8: Receiver circuit

6.2. Control system implementation

6.2.1. Position control system

FPGA is a generic computational platform, with parallel processing and runtime reconfiguration capabilities. Linear control system algorithms are generally multiply-accumulate intensive functions and 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, the two-degrees-of-freedom controller is implemented in an FPGA. The controller is designed for angle position control of a servomotor. The controller consists of three main parts: 1) an integer and floating point number transform unit, 2) a state variable computing unit and 3) an output computing unit. These are shown in Figure 6-9.

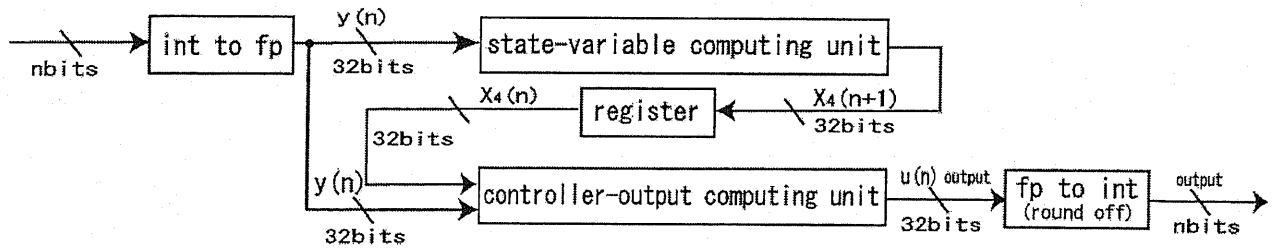


Figure 6-9: Block diagram of the controller in FPGA

The data format for storing floating point numbers in the controller is the IEEE 754 [45] standard. Single precision floating point uses the 32-bit IEEE 754 format shown in Figure 6-10. 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 [44], [46].

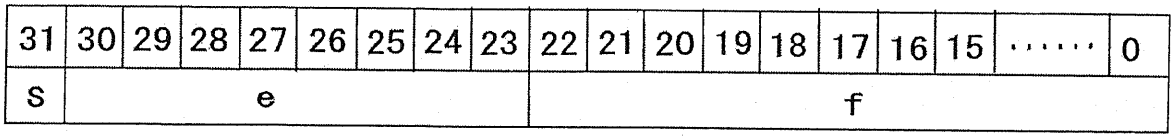


Figure 6-10: 32-bit floating-point number format.

Floating point multiplication is very similar to 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.

In addition, a DA converter and a counter are used, therefore, it was necessary to design a way to transform single precision floating point numbers to integers and integers to single precision floating point numbers [44]. In the first conversion step of floating point numbers to integers, the value of right shift is computed. The second step shifts the mantissa to right. Then, the mantissa is adjusted and rounded off to output. To convert from integers, the mantissa of the input data is separated, and an exponent is built. The exponent is then adjusted, and the mantissa is output.

6.2.2. Current control system

In the previous section, the FPGA solution for the proposed system based on the servo amplifier was completed. In this section, the current control loop of servo motor is implemented in FPGA, and the circuit is described in VHSIC hardware description language (VHDL). FPGAs con be considered as an appropriate solution in order to boost performances of controllers and consequently to reduce the gap between the analog and digital world [29]. As a result, FPGAs are quite mature for the electrical drive applications. They have already been applied with success to the control of PWM converters, machine drives. Hence, to improve the performance of the proposed system, the FPGA solution for minor current control loop of servo motor is proposed. The design of the current control system, based on floating point number level, is presented. The structure of the designed current control system circuit in FPGA is shown in Figure 6-11. This architecture is divided into several units: 1) the motor counter unit, 2) the trigonometric function generator unit, 3) the AD converter interface unit, 4) the axis converter unit, 5) the integer and floating point number transform unit, 6) the speed computation unit, 7) the deadbeat controller module unit, 8) the space vector sector selection unit, 9) the switch firing time computation unit and 10) the PWM generator unit.

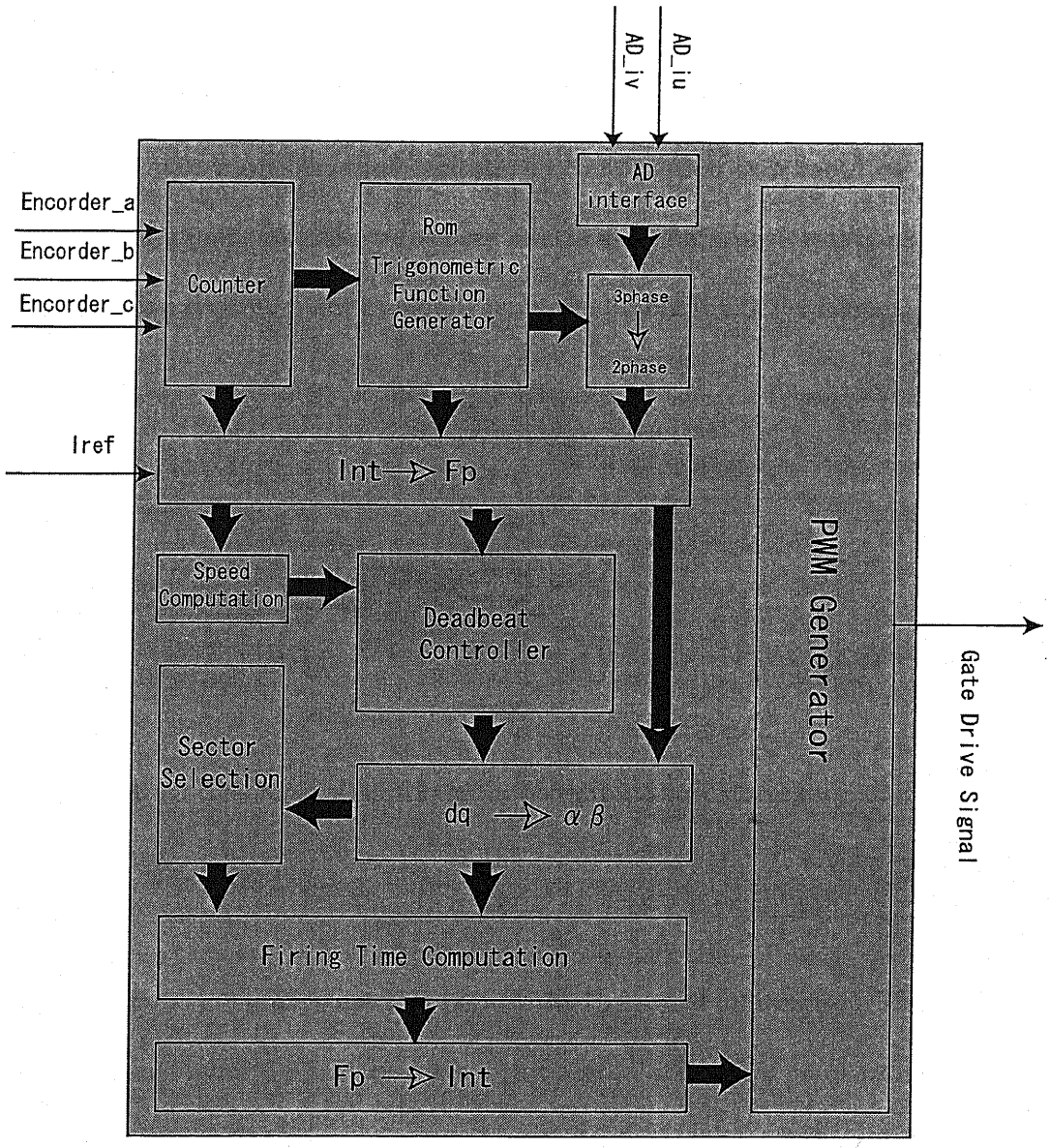


Figure 6-11 Structure of the designed circuit.

The counter unit receives and counts the motor-position signals. In the previous section, the structure of the unit and the processing of the unit were expressed.

The trigonometric function generator unit and the coordinate transformation unit consist of a Rom block and the integer operator, as shown in Figure 6-12. The 12-bit address of the Rom is generated by the counter unit, and the data width of the trigonometric wave is set to 10 bits. Before the coordinate transformation module, all computation was based on integer level. The integer and floating point number transform unit is used for transformations between integers and single precision floating point numbers using the 32-bit IEEE 754 format.

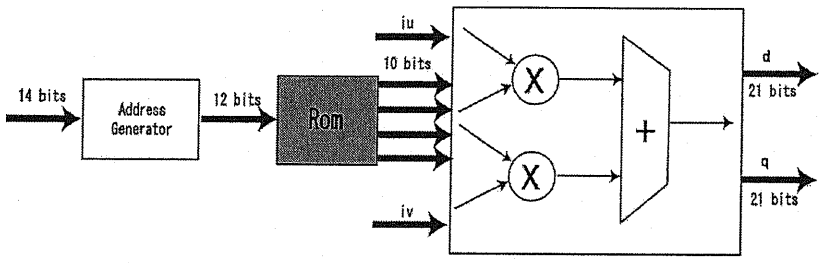


Figure 6-12: The block diagram of the coordinate transformation unit circuit.

The speed computation unit is used for measuring the speed information. The structure of the circuit is shown in Figure 6-13. It consists of a counter, a floating point number transform, a floating point number divisor, comparators and a direction determining circuit. This unit refreshes the speed information when the encoder pulse is rising. The speed computing equation is

$$\omega = \frac{2\pi}{PnT} \tag{6.1}$$

where P is the pulse number per round rising by the encoder of the motor and n is the clock

counting number between the neighboring pulses of the encoder.

The deadbeat controller module unit and switch firing time computation unit consist of the arithmetic logic unit, which includes floating point addition/subtraction and floating point multiplication. It consists of floating point multiplication, an absoluter, a comparator and a six-to-one MUX unit. With this unit, we can accurately obtain the vector sector.

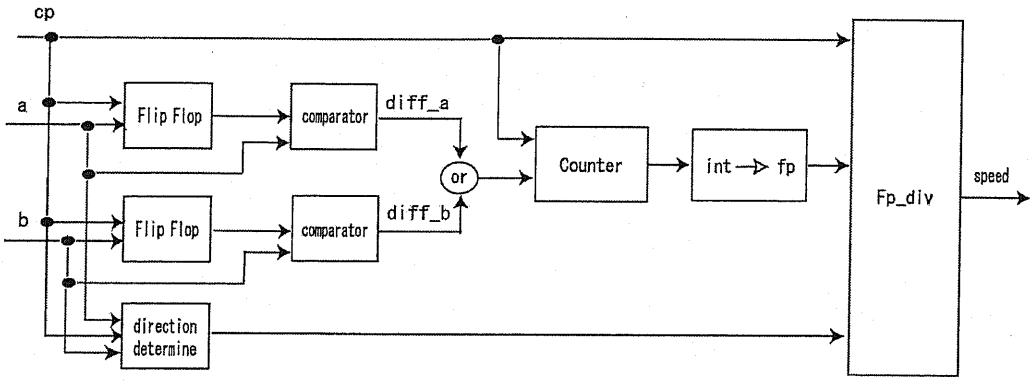


Figure 6-13: The block diagram of the speed computation unit circuit.

With the current control module implementing in FPGA, the calculating time is shorted obviously. The clock frequency of FPGA is 50MHz and the total calculation time is 1.82μs, which takes 91 clocks. The first 42 clocks the current information and the position information is achieved. From clock 42 to clock 51 the axis conversion is completed. From clock 51 to clock 63 the deadbeat control module is completed and then from clock 63 to clock 91 the firing time of each switch is computing completely. Figure 6-14 shows the circuit for the timing chart.

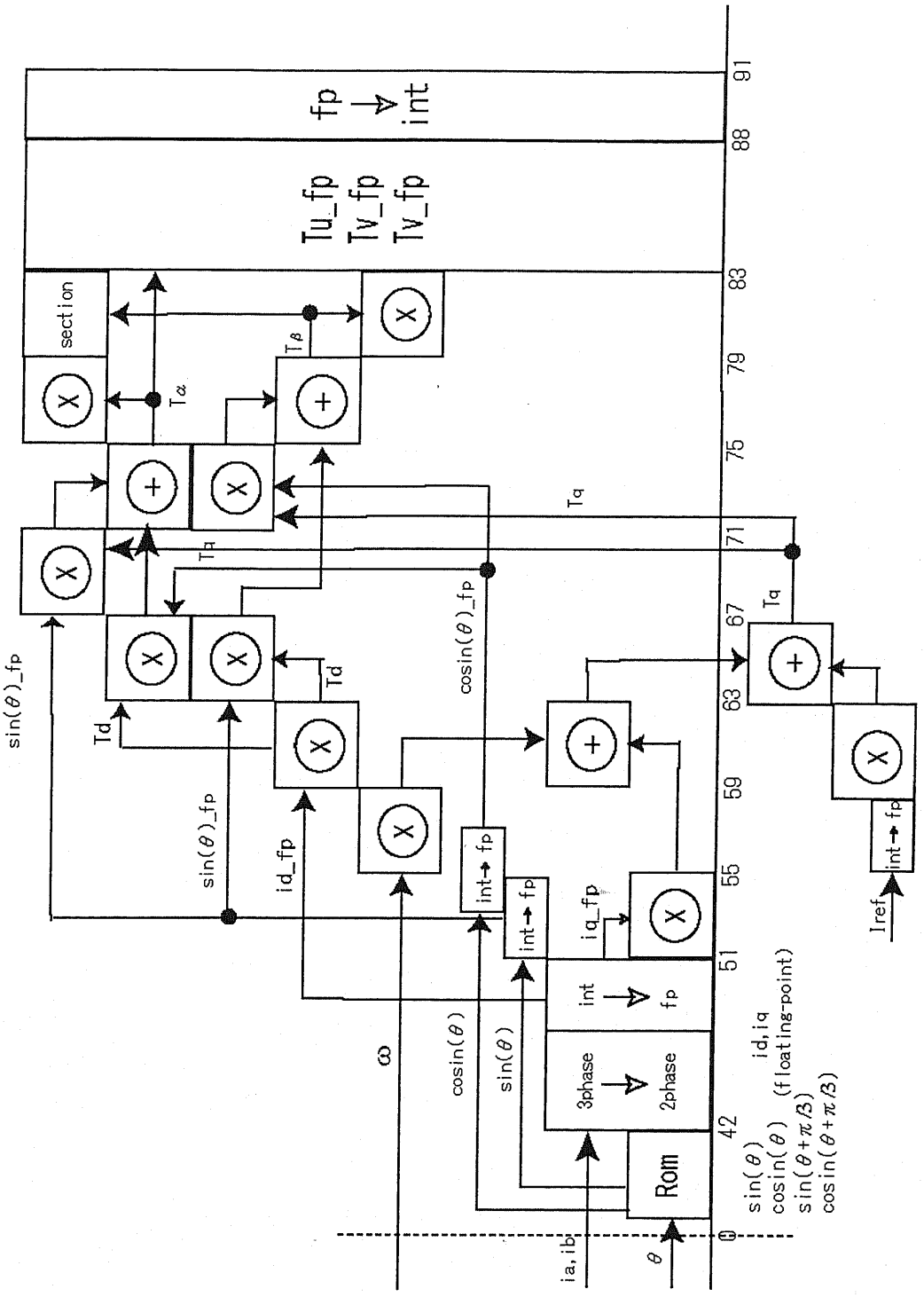


Figure 6-14: Timing chart of the current controller circuit for FPGA

Chapter 7

7 Experiment using Servo-amp

7.1. Experimental setup

Experiments were performed to illustrate the performance of the proposed system. Here, the AC motor current control loop is employed by the servo amplifier. A simple peer-to-peer networked motor control system was set up as shown in the block diagram in Figure 7-1. The system is implemented in two Altera Stratix EP1S10F780C7ES FPGAs [47]. The basic FPGA clock frequency is 33.33 MHz. As shown by the results of logic synthesis and fitting using Altera Quartus II [48], the controller circuits take up approximately 50% of the total FPGA logic elements. The two-degrees-of-freedom controller takes 20 base clock (1 clock = 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 amplifier (Yaskawa Electric Co.) and an AC servomotor (Yaskawa Electric Co.) represent the controlled plant. The servo amplifier 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.

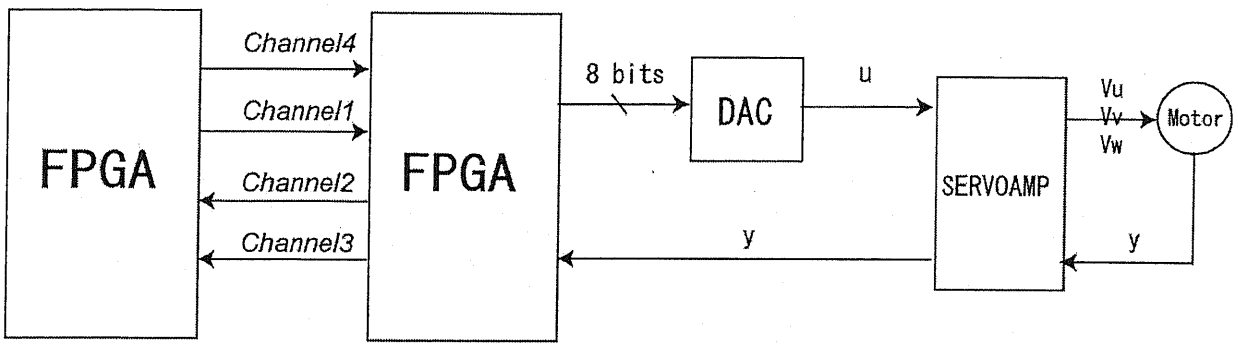


Figure 7-1: The block diagram of the experimental setup.

7.2. Experimental results

7.2.1. System response without noise

Different response time constants ($\tau = 0.05$ s and $\tau = 0.2$ s) is set to verify the response performance of the proposed system with step position references of 1 rad and 0.1 rad respectively. Figure 7-2 and Figure 7-3 show the position output response of the motor without noise. By comparing this with the simulation results given in Figure 7-2 and Figure 7-3, we can confirm a reliable system response.

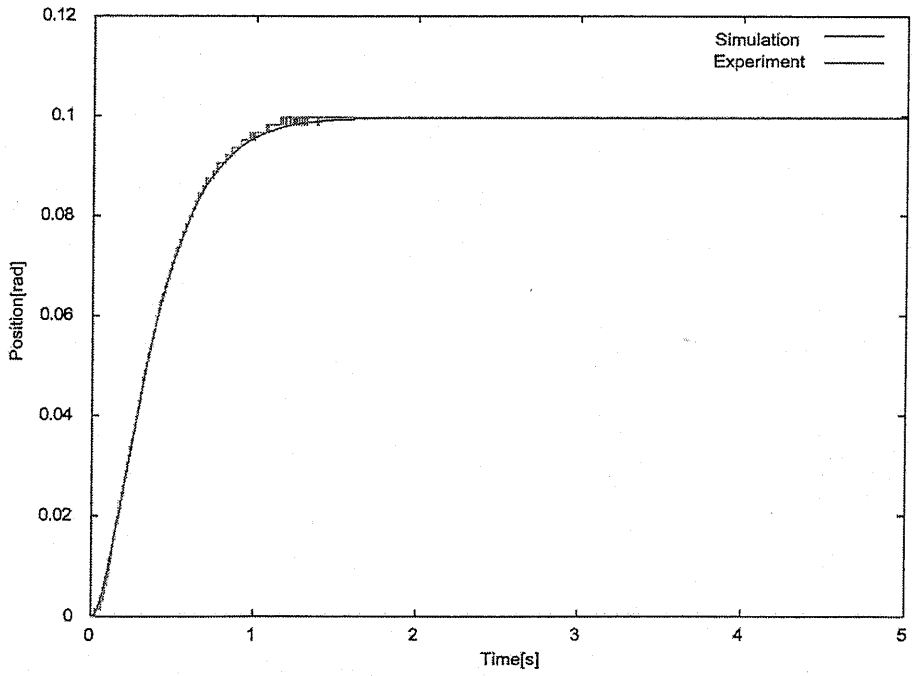


Figure 7-2: Control system response with high-speed communication links without noise (ref = 0.1 [rad], $\tau = 0.2$ [s])

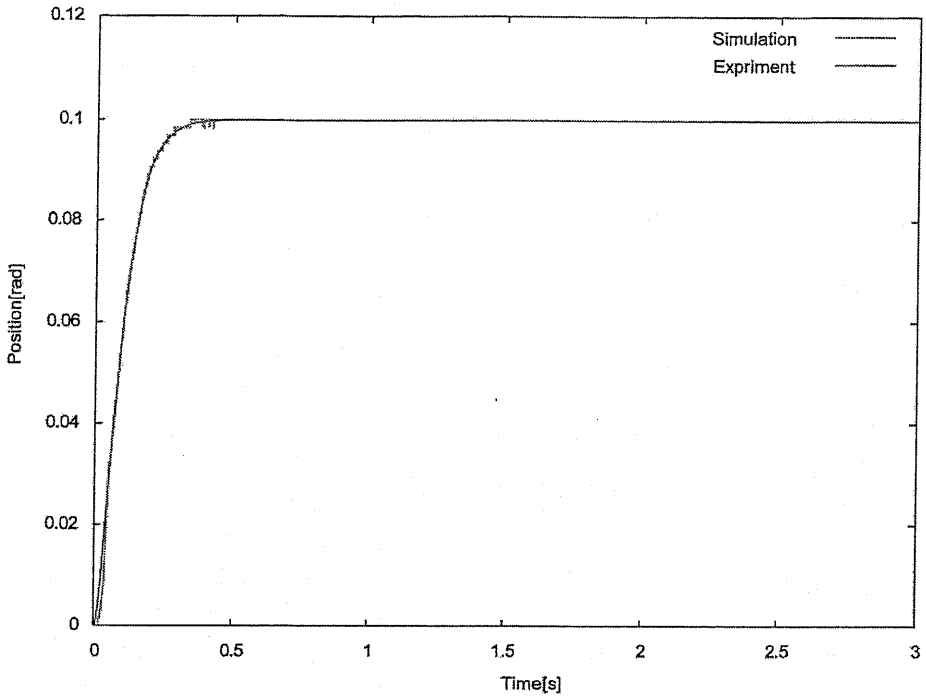


Figure 7-3: Control system response with high-speed communication links without noise.

(ref = 0.1 [rad], τ = 0.05 [s])

7.2.2. System response with noise

In the motor system, the largest source of noise is electrical fast transient (EFT), which is coupled by the changing in the magnetic field around the communication channels. Linear current-induced magnetic field changing to couple the EFT noise was adopted in [57]. In the paper, the EFT noise is coupled by the inductance current-induced magnetic field changing. In theory, the noise coupling method of linear current-induced magnetic field changing and inductance current-induced magnetic field changing are the same. The noise generator circuit is shown in Figure 7-4. The noise frequency is 10 kHz, and a bit error of 0.5478% is realized. The position responses of the motor with the noise frequency set at 30 kHz (bit error = 0.9234%) and 40 kHz (bit error = 1.5305%) are shown in Figure 7-5 and Figure 7-6 respectively. Figure 7-7 shows that the system response is unstable and unreliable with a 1.5305% bit error. The simulation is tolerated with a 3% bit error; however, in the experiment, the channel noise also affects synchronization among FPGAs, an element that cannot appear in the simulations. From the experimental results, we can confirm the proposed system can provide suitable and reliable position control performance and correctly compensate for noise, under a bit error of around 0.9%.

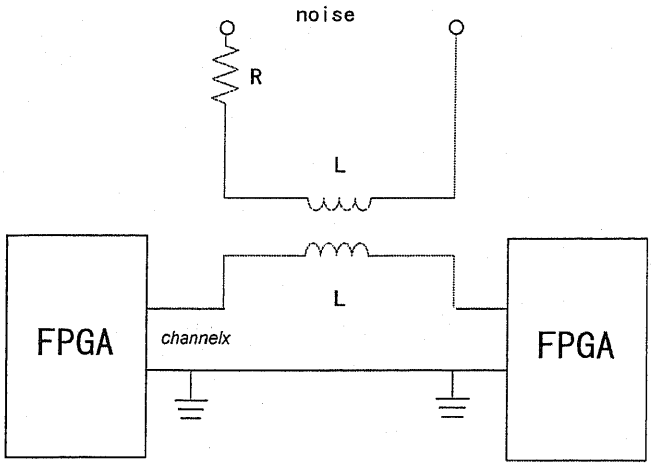


Figure 7-4: Noise-generator circuit.

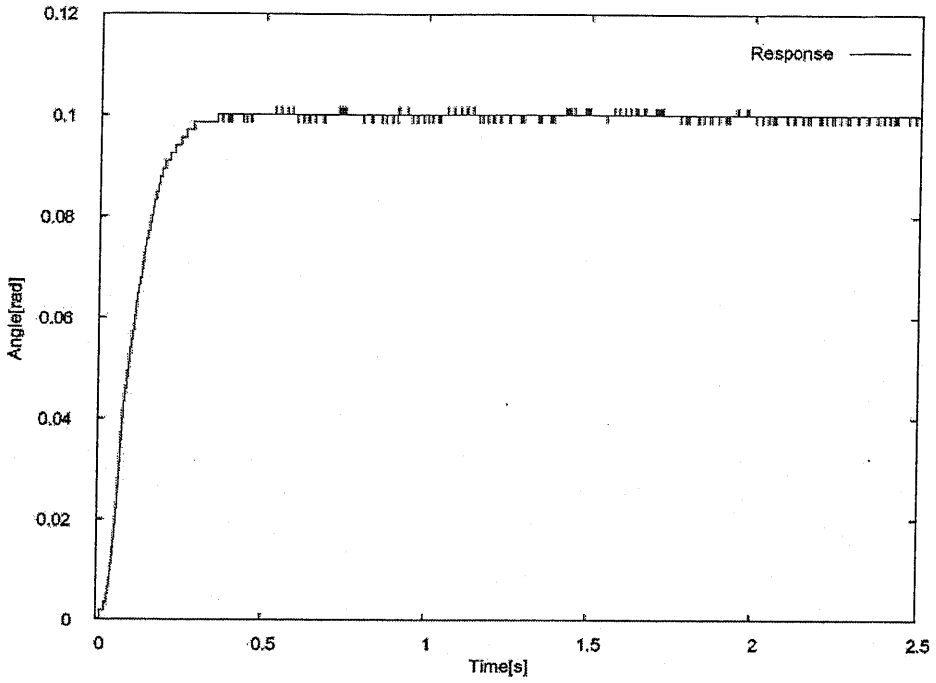


Figure 7-5: Experimental result with noise over channels (0.5478% bit error).

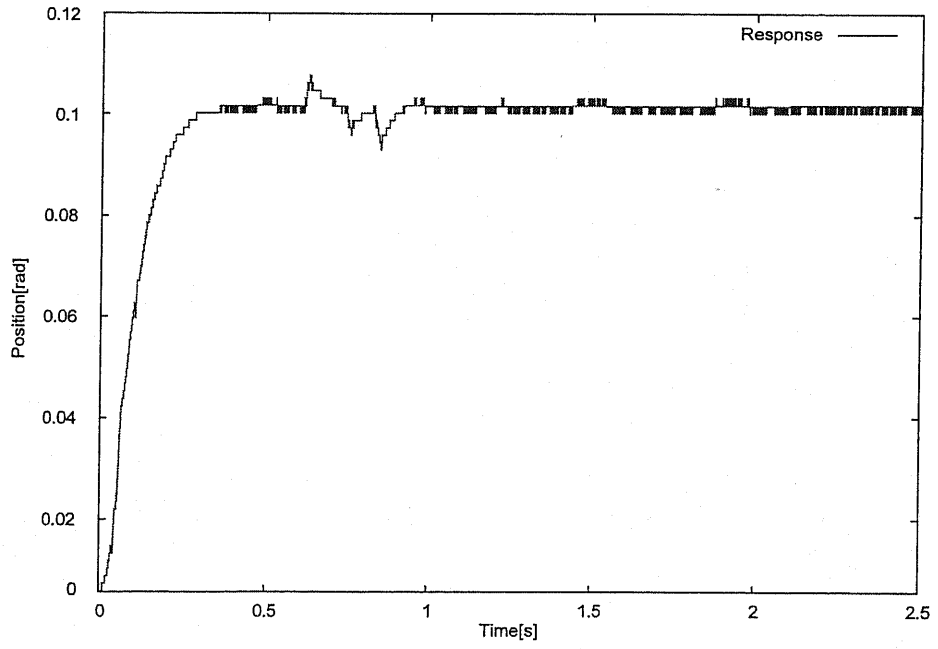


Figure 7-6: Experimental result with noise over channels (0.9234% bit error).

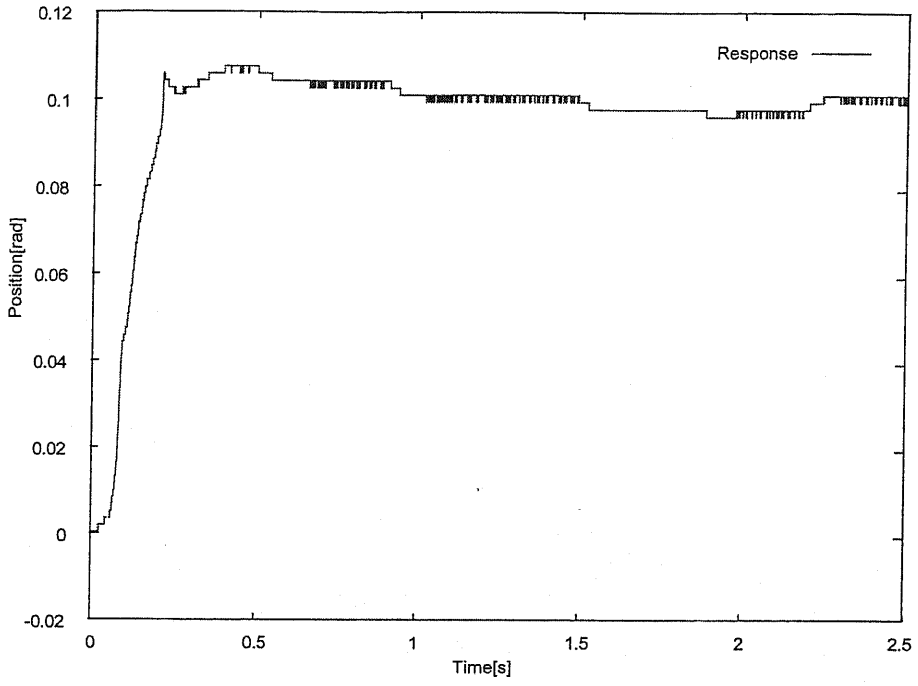


Figure 7-7: Experimental result with noise over channels (1.5305% bit error).

Chapter 8

8 Experiment Using Proposed Current Control Loop

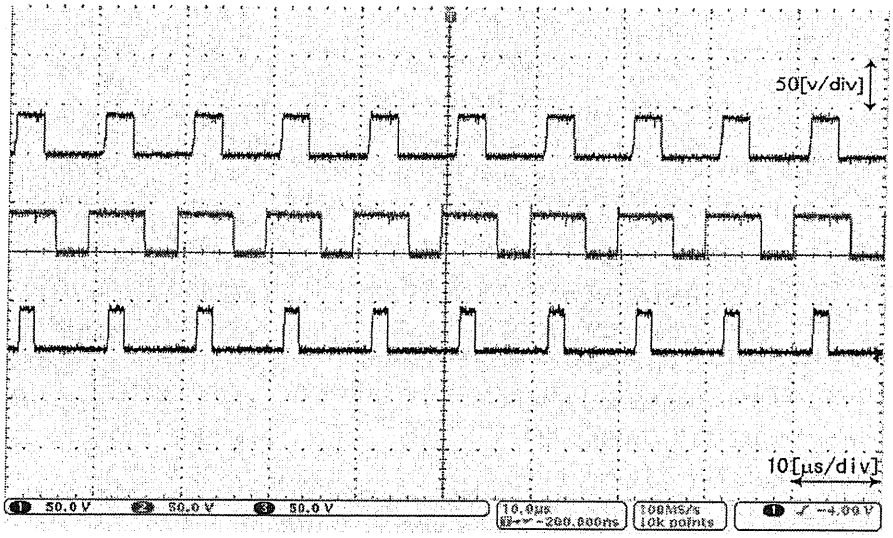
In chapter seven, the experiments illustrated the performance of the proposed system by using servo amplifier. In this chapter, the experiments are performed to illustrate the performance of the proposed system by using the FPGA solution for minor current control loop instead of servo amplifier.

8.1. Experimental results for current control loop

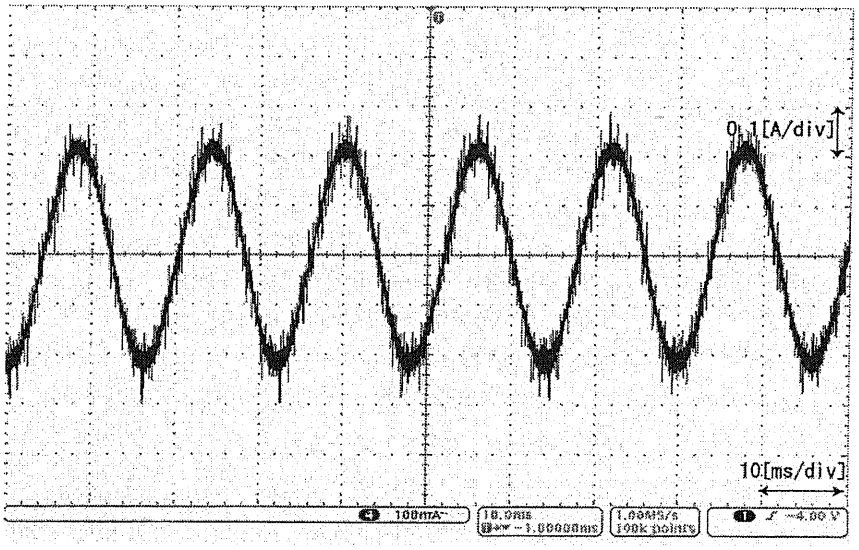
In order to realize the proposed system, the MOSFET switch PWM inverter was designed for a carrier frequency of 100 kHz. The three phase inverter driver chip employed was the IR21094, working on a 600 ns dead time level, and the MOSFET switch employed was the IR634. The inverter firing pattern was generated by the FPGA. The three phase RL load was set up in the experiment. From Figure 8-1, we can confirm that the inverter functions correctly for a 100 kHz carrier frequency and 32.5 Hz sine wave reference.

The proposed FPGA-based current control system for the AC servo system was set up as shown in the block diagram of Figure 8-2. The system was implemented in Altera CycloneII EP2C70 FPGA. The computation time of the current controller architecture was 1 μs. By adding the AD conversion time, the entire execution time was 1.82 μs. The AD converter employed was the eight-bit AD7821. In the experimental setup, the DC supply of the PWM inverter was 30 V,

and the carrier frequency was 100 kHz. The experimental results are shown in Figure 8-3 with the step references 0.125A, 0.25A and 0.5A. From the results, we can confirm that the FPGA-based current control system can produce the desired performance.



(a)



(b)

Figure 8-1: (a) Switching pulse for 100 kHz carrier frequency. (b) Current waveforms 100 kHz carrier frequency.

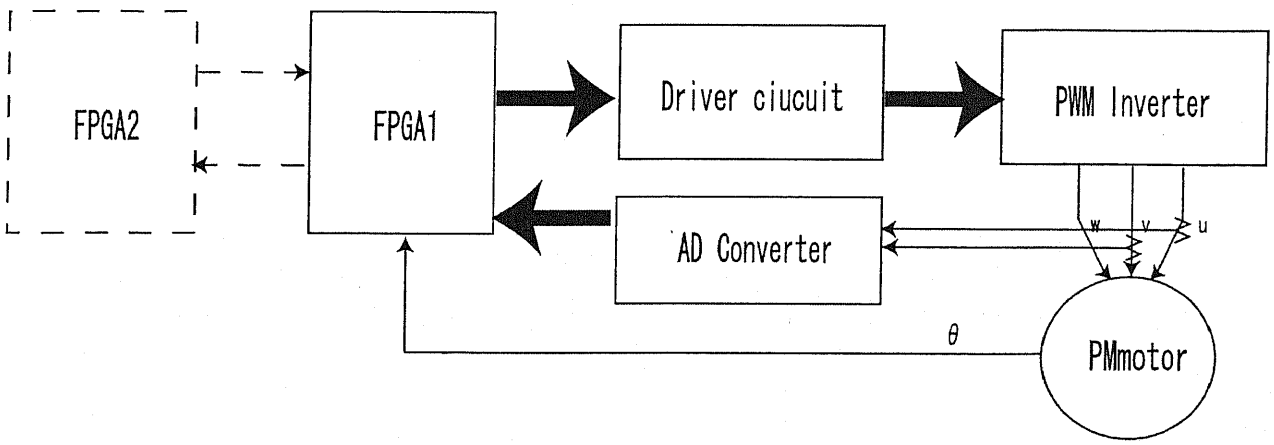


Figure 8-2: Block diagram of the experimental setup.

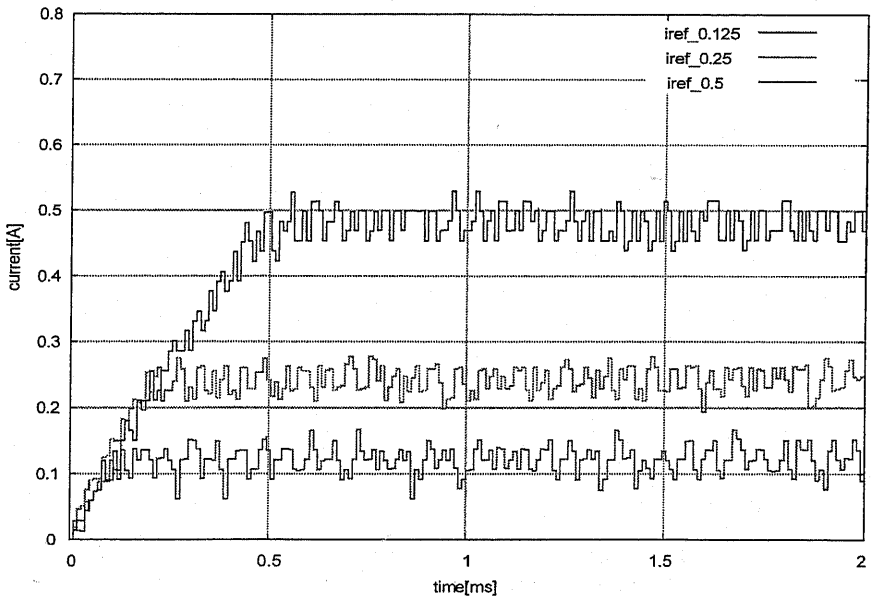


Figure 8-3: Experimental results for current responses of q-axis

8.2. Experimental results for proposed networked control system

In the section, experiments were performed to illustrate the performance of the proposed networked control system. A simple peer-to-peer networked motor control system was set up as shown in the block diagram in Figure 8-4. The system was implemented in two Altera CycloneII EP2C70 FPGAs. The FPGA implementation for the position control system and communication system were reported in chapter six. The basic FPGA clock frequency was 50 MHz. The results of logic synthesis and fitting were obtained using Altera QuartusII. The FPGA-based communication system took 20 ns to transmit one bit in serial communication. To read the transmitted data accurately, a 5x oversampling method was utilized to provide more detailed sampling periods. The data rate is 50 Mbps per channel. The entire system timing chart is shown in Figure 8-5. The position controller execution time was 400 ns. In combination with the communication system, the execution time was 1 μs. In combination with the carrier frequency of the PWM inverter in the current control system, a 100 kHz networked control system was realized. The different response time constants ($\tau = 0.05$ s and $\tau = 0.02$ s) is set to verify the response performance of the proposed system with the step position reference of 1 rad. Figure 8-6 shows the position output response of the motor. Figure 8-7 shows the responses of the motor for references 0.1 rad and 0.05 rad with the response time constants $\tau = 0.005$. From the experimental results, we can confirm the proposed system can provide suitable and reliable control performance. Figure 8-8 shows the responses of the motor and the current of q-axis for references 0.05 rad with the response time constants $\tau = 0.0015$. Overshoot is occurred in this result. From the current response we can see that the friction affects the response performance.

For much smaller position reference, the resolution of AD convertor is also limited the response performance.

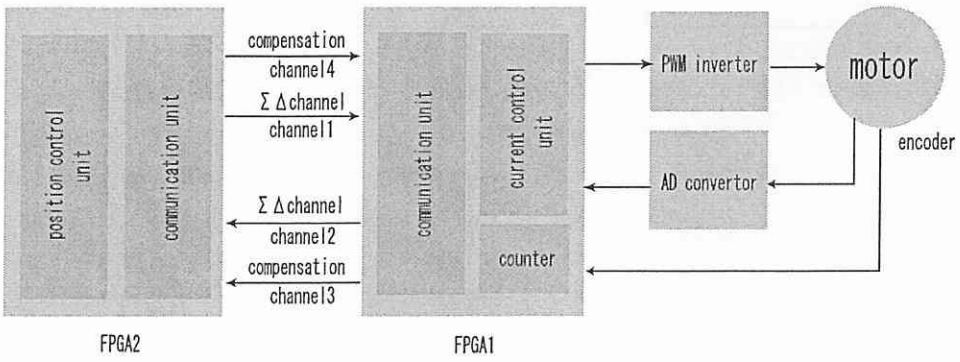


Figure 8-4: Block diagram of the experimental setup

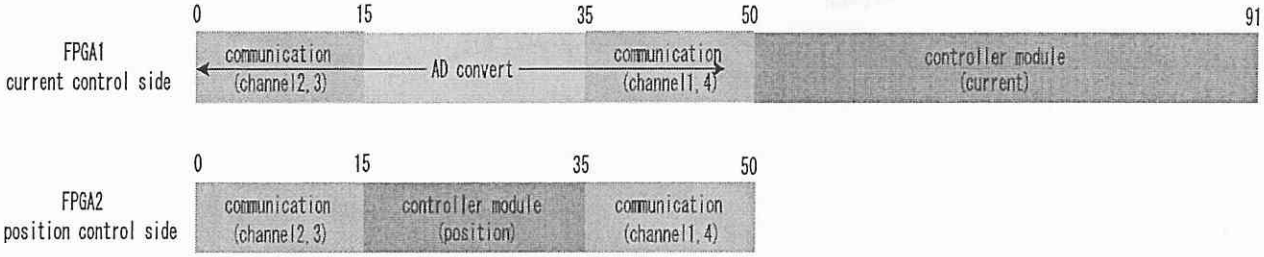


Figure 8-5: Timing chart for proposed networked control system

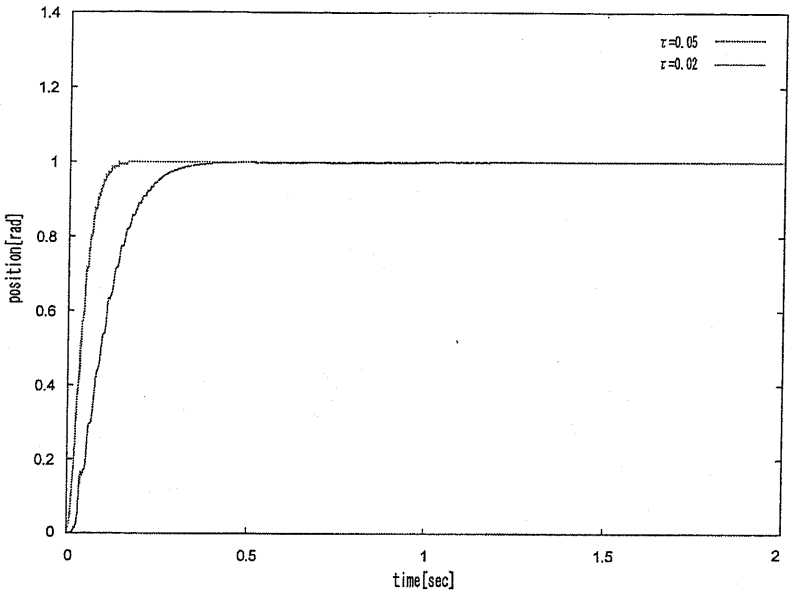


Figure 8-6: Proposed system response with high-speed communication links.

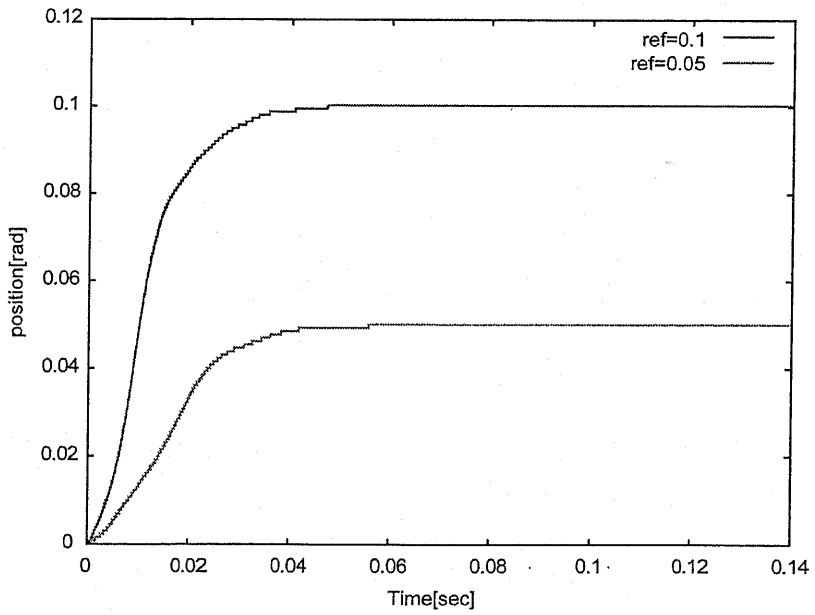


Figure 8-7: Proposed system response with high-speed communication links ($\tau = 0.005$)

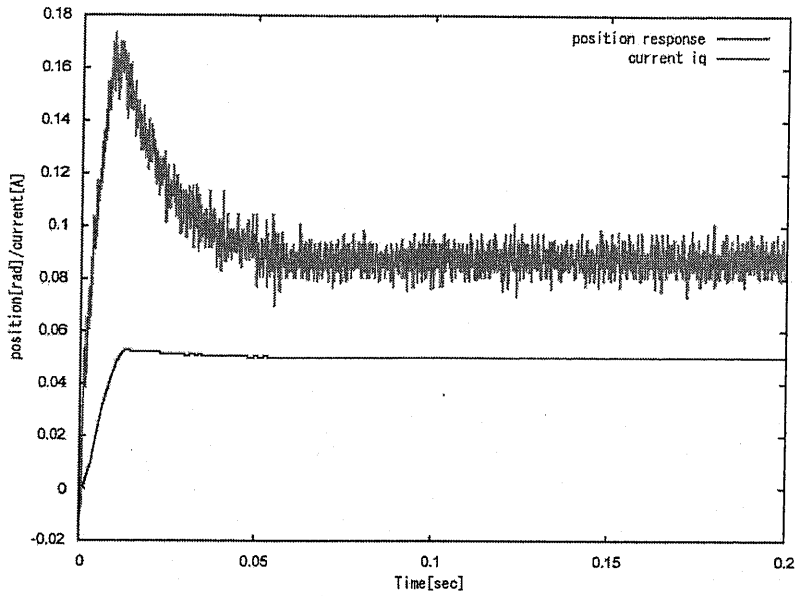


Figure 8-8: Proposed system response with high-speed communication links ($\tau = 0.0015$)

Chapter 9
9 Conclusion

Networked control systems are becoming increasingly important as industries focus on distributed computing in automation. As computing costs decrease and computing power increases, industry has relied more on distributed computers to deliver efficiency and increased yield to production lines. For the industrial requirement, the system must combine safety with real-time demand. In real-time networked control systems, the real-time requirement is much more difficult to obtain for motion control systems than with other applications. For effective high performance motion control, demand data and feedback data must be transmitted synchronously, with very short cycle times and low latency. Any transmission delay represents a phase delay in the control system, which limits the achievable gain and the effective response time of a machine. For the existing networked control system, it is difficult to shorten the sampling period because of the limited data rate. Hence, a high speed networked control system using sigma-delta modulation for the servo system was proposed.

The first part of this paper presented the system configuration using sigma-delta modulation. In a communication system, sigma-delta modulation is adopted to realize high speed data transmission and data compression. In this way, it can meet the requirements of the control and communication system for high speed sampling and real time. Next, the novel noise compensation scheme was presented. The noise compensation scheme depended on the designed controller and the noise compensation channel. Due to this noise compensation scheme, the

system can be satisfied with the requirement of the control system, and the noise compensation scheme can achieve much higher real-time performance than the existing system that uses, for example, CRC noise detection. The simulation results were also given to demonstrate the performance. In the middle part of the paper, an FPGA solution for the proposed system was developed. The FPGA circuits and the controller computing process were described. Finally, the proposed high speed system was implemented with a 0.6-μs sampling period position control system using a servo amplifier; a 10-μs cycle time networked control system was realized in the experiments, in which the current control loop of motor was implemented in FPGA. In the current control system, the 100 kHz PWM inverter was designed. The total execution time of the controller and communication was 1.82 μs. The system performance was demonstrated by numerical experimental results. The abovementioned features of the proposed system allow the achievement of a much higher real-time performance than the existing real-time networked control system

Reference

[1] P. Ogren, E.Fiorelli, and N. E. Leonard, "Cooperative control of mobile sensor networks: Adaptive gradient climbing in a distributed environment," *IEEE Trans. on Automat. Contr.,* vol. 49, no. 8, pp. 1292-1302, 2004.

[2] C. Meng, T. Wang, W. Chou, s. Luan, Y.Zhang, and Z. Tian, "Remote surgery case: Robot-assisted teleneurosurgery," in *IEEE Int. Conf. Robot. and Auto. (ICRA'04),* vol. 1, pp. 819-823, 2004.

[3] P. Seiler and R. Sengupta, "Analysis of communication losses in vehicle control problems," in *Proc. 2001 Amer. Contr. Conf.,* vol. 2, pp. 1491-1496, 2001.

[4] 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.

[5] 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.

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

[7] Bosch, Postfach 50, "D-700 Stuttgart 1. CAN Specification, version 2.0 edition," 1991.

[8] Felser, M. Sauter, T. "Standardization of industrial Ethernet - the next battlefield?" *Factory Communication Systems, 2004. Proc. 2004 IEEE Int. Workshop on*, pp. 413- 420, 2004.

[9] J. Wilkinson, "Distributed motion technology goes PC," Industrial Ethernet Book, Issue 21.

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

[11] 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.

[12] 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.

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

[14] 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.

[15] 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.

[16] 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.

[17] 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.

[18] E. Monmasson and M. Cirstea, "FPGA design methodology for industrial control systems-a review," *IEEE Trans. on Industrial Electronics,* vol. 54, no. 4, pp. 1824-1842, 2007.

[19] 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.

[20] 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.

[21] 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.

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

[23] 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.

[24] 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.

[25] E. Kappos and D.J. Kinniment, "Application-specific processor architectures for embedded control: case studies," *Microprocessors and Microsystems*, pp. 225-232, 1996.

[26] 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.

[27] 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.

[28] 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.

[29] 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.

[30] 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.

[31] 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.

[32] 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.

[33] 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.

[34] M. N. Cirstea, and A. Dinu, "A VHDL holistic modeling approach and FPGA implementation of digital sensorless induction motor control scheme," *IEEE Trans. on Industrial Electronics*, vol. 54, no. 4, pp. 1853-1864, 2007.

[35] 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.

[36] 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.

[37] 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.

[38] 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.

[39] 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.

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

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

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

[43] 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.

[44] 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.

[45] "IEEE standard for binary floating-point arithmetic", ANSI/IEEE Std., IEEE 754-1985, 1985.

[46] 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.

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

[48] Altera, Inc., *Quartus Iversion 5.0 handbook*, San Jose, CA, 2005.

[49] Paula Doyle, "Introduction to real-time Ethernet 1," *A Technical Supplement to Control Network*, Vol. 5, No. 3, pp.1-4, 2004

[50] John C. Eidson and Bruce Hamilton, "IEEE-1588 Node synchronization improvement by high stability oscillators," *Workshop on IEEE-1588, Standard for a Precision Clock Synchronization Protocol for Networked Measurement and Control Systems*, pp. 95-101, 2003

[51] Bob Lounsbury, "Surviving industry", *The Industrial Ethernet book*, Issue 7, September 2001

[52] Rahmani, M.Muller-Rathgeber, B and Steinbach, E. "Error detection capabilities of automotive network technologies and Ethernet - a comparative study", *Intelligent Vehicles Symposium, 2007 IEEE*, pp.674-679, June 2007

[53] C. Canudas-de-Wit, Rubio, F.R. Fornes, J. Gomez-Estern, F.," Differential coding in networked controlled linear systems," *Amer. cont. conf. 2006,* Minnesota. June 2006.

[54] K. Sakata and H. Fujimoto, "Perfect tracking control of servo motor based on precise model considering current loop and PWM hold," *IEEJ Trans. on Ind. Appl.* vol.127, no. 6, pp.587-593, 2007.

[55] Felser. M, "Real-time Ethernet-industry prospective," *Proceedings of the IEEE*, Vol.93, No.6, pp.1118-1129, 2005.

[56] Jean-P. Thomesse, "Fieldbus technology in industrial automation," *Proceedings of the IEEE*, Vol.93, No.6, pp. 1073-1101, 2005.

[57] "Comparison of Cat.5e to Cat.6 using EFT pulse EMI," *Nexans data communications competence center*, July 12, 2007, http://www.nexans.com/US/files/DCCC03040901R1.pdf.

[58] German Standards 19245-1 to 19245-3, PROFIBUS, Process Fieldbus, 1990.

[59] Danish Standard, DS 21906. P-Net, Multimaster, MultiNet Fieldbus for Sensor, Actuator and Controller Communications.

[60] Real-Time Ethernet: SERCOS III: Proposed for a Publicly Available Specification for Real-Time Ethernet, Doc. IEC 65C/358/NP, 2004.

Related Publications

Journals

[1]. Tianjian Li and Yasutaka Fujimoto, "Control system with high-speed and real-time communication links", *IEEE Trans. on Industrial Electronics*, vol. 55, no. 4, pp. 1548-1557, 2008.4

[2]. Tianjian Li and Yasutaka Fujimoto, "FPGA-Based real-time and high-speed servo control system with communication channels," *IEEE Trans. on Industrial Electronics*. (submitted)

International Conference

[1]. Tianjian Li and Yasutaka Fujimoto, "FPGA-Based current controller for high-speed communication and real-time control system", *Proc. IEEE Industrial Electronics Society Annual Conference (IECON)*, pp. 210-215, 2008.11.

[2]. Tianjian Li and Yasutaka Fujimoto, "On a control system with high-speed and real-time communication", *Proc. IEEE Int. Workshop on Advanced Motion Control (AMC)*, vol. 1, pp. 12-17, Istanbul, 2006.3

Domestic Conference

[1]. Tianjian Li and Yasutaka Fujimoto, "High-speed current control system of servo motor based on communication-control system by using FPGA", *Proc. Japan Industry Applications Society Conference (JIASC'08)*, 2-74, 2008.8. (in Japanese)

[2]. Tianjian Li and Yasutaka Fujimoto, "High-speed and Real-Time networked servo control

system considering current feedback ", *Proc. Japan Industry Applications Society Conference (JIASC'07)*, 2-66, 2007.8. (in Japanese)

[3]. Tianjian Li and Yasutaka Fujimoto, "High-speed networked control system using sigma-delta modulation", *IEEJ Technical Meeting on Industrial Instrumentation and Control*, IIC-09-47, 2009.3. (in Japanese)

[4]. Tianjian Li and Yasutaka Fujimoto, "High-speed and Real-time communication control system based on current control ", *IEEJ Technical Meeting on Industrial Instrumentation and Control*, IIC-07-47, 2007.3. (in Japanese)

[5]. Tianjian Li and Yasutaka Fujimoto, "A study of high-speed and real-time communication control system with FPGA", *IEEJ Technical Meeting on Industrial Instrumentation and Control*, IIC-06-60, pp. 87-91, 2006.3. (in Japanese)

[6]. Tianjian Li and Yasutaka Fujimoto, "The basic research of high-speed and real-time communication control system", *IEEJ Technical Meeting on Industrial Instrumentation and Control*, IIC-05-51, pp. 67-72, 2005.3. (in Japanese)