A Closed-Loop Stepper Motor Drive Based on EtherCAT

Vinh Quang Nguyen[1], Nhan Van Phan Tran[1], Hoang Ngoc Tran[1], Kien Minh Le[1], and Jae Wook Jeon[2]
[1]Department of Electrical and Computer Engineering, Sungkyunkwan University, Suwon, Korea
{ngqvinh, phannhan, hoang92, kienha}@skku.edu
[2]School of Information and Computer Engineering, Sungkyunkwan University, Suwon, Korea
jwjeon@yurim.skku.ac.kr

***Abstract*—This paper presents the design and implementation of a motion control system using EtherCAT. The EtherCAT frame with an embedded motion profile is introduced. Additionally, a closed loop stepper motor that uses the embedded data of the EtherCAT frame is shown. For position control purposes, a lead angle estimation was designed. The Proportional Integral Feed Forward (PI-FF) controller was implemented to overcome the problem with reverse motor rotation. Finally, some experimental results are shown to verify the performance of the EtherCAT based motor drive.**

Keywords*—*real-time Ethernet; EtherCAT; stepper motor; closed loop controller; lead angle ; feed forward control

I. Introduction

Stepper motors are widely used in motion control systems that require easy positioning due to their structure and control simplicity [1]. Therefore, studying and increasing the performance of stepper motors is vital to engineering. A stepping motor can be controlled by using an open-loop controller without electrical signal feedback from an encoder [2]. With this method, the step rotor follows the pulses that are produced by the controller through pulse width modulation (PWM). When the motor runs in a resonance condition, torque disturbance, the motor can not synchronize with the motion command. Therefore, various studies were conducted to increase the performance of stepper motors. Grimbleby [3] proposed a simple closed loop for a stepping motor that is better than open-loop control when using signal feedback from an encoder and the closed-loop remains synchronized. Stuart A. Schweid [4] demonstrated a closed-loop controller that combined torque compensation with a current reference to overcome the changed torque and support smooth velocity. Quy Ngoc Le [6] proposed a speed estimation based on electrical torque for a smooth low speed stepper motor. A low speed damping controller for a step motor based on a neural network is introduced in [8]. Additionally, an application of velocity profile generation is introduced for exact positioning in [5]. Jung Uk Cho [7] also presented a method for designing a multiple axis motion control chip based on a field-programmable gate array (FPGA).

In motion control systems, multiple motor drives are connected by a network in order to save space and money. Therefore, the real-time network gradually becomes an important element of transfer data, security, and synchronism. CAN has been used in automation networks for a long time. Byung Yun Oh [9] presented a motion controller using CAN based on an FPGA. The authors used CAN King for the master program software to transfer the motion profile data. An overview of a CAN based motion system is also given. However, CAN has some problems with bandwidth, payload and event-triggers. The CAN-FD (CAN-flexible data rate) network supplies a higher payload and the baud rate is analyzed in [10]. It is an idea choice for overcoming the existing problems in CAN. However, the CAN FD transceiver hardware has not appeared on the market yet. A gateway between the CAN and CAN-FD based on FlexRay-backbone is presented in [11]. In order to overcome the existing problems, real-time Ethernet has become one of the best choices in industrial networks. In addition, in some networks such as Mechatrolink, Device Net, and EtherNet/IP, EtherCAT plays a vital role in real time Ethernet. With an on the fly mechanism, high bandwidth, and the ability to support multiple devices in one frame, EtherCAT can transfer data to multiple drives in a short period of time by using just one frame. Max Felser [12], and Kun Ji [13] introduced the concept of networked control systems via Ethernet. The communication principle of EtherCAT is presented by the EtherCAT Technology Group [14]. Studies on motion control using EtherCAT were introduced by Jin Ho Kim [15], Vinh Quang Nguyen [16], and Do Eon Lee [17]. The authors in these papers only provide an overview of motion control systems, the details and process for improving the motion control system using EtherCAT have not been presented. In this paper, we present the design and implementation of a motion control system using EtherCAT. The embedded EtherCAT frame is implemented. Additionally, the extracted motion data from the EtherCAT frame is explained. Furthermore, a closed-loop motor drive that includes a lead angle estimator and a Proportional Integral Feed-Forward (PI-FF) controller is also developed.

The remainder of this paper is organized as follows. In Section II, the EtherCAT communication and the mapping and un-mapping of the data frame from EtherCAT are introduced. In Section III, the lead angle estimation is presented. Section IV details our implementation in a closed-loop stepping motor drive. Experimental results are performed to verify the performance of the EtherCAT-based motion system in Section V and Section VI concludes this paper.

II. EMBEDDED ETHERCAT TELEGRAM

A. *EtherCAT frame overview*

Real-time Ethernet has gradually started playing an important role in filebus-based motion control system. In addition to

978-1-5386-1127-2/17/$31.00 ©2017 IEEE

several protocols that have been developed such as SERCOS, CC Link, and Mechatrolink, EtherCAT is known as the best choice for high bandwidth, short cycle time, and high time-synchronism. An overview of the EtherCAT frame is shown in Fig. 1. The EtherCAT telegrams are embedded into the payload of the Ethernet frame. With an ”on-the-fly” mechanism and distributed clock (DC) method, EtherCAT can support multiple drives with a reduced time cycle and constant node delay.

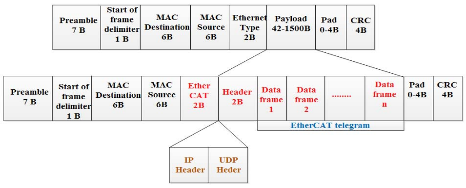

Fig. 1. EtherCAT Frame

B. *DETAILS ABOUT ETHERCAT TELEGRAM*

EtherCAT uses a mailbox standard to exchange parameter data and configure the system in the pre-operational state such standards include Ethernet over EtherCAT (EoE), CANopen over EtherCAT (CoE), and File Access over EtherCAT. In order to control and support the large number of stepper motors in different factories, we used CoE and the CiA402 Drive Profile for mailbox implementation. With this method, the service data object (SDO), process data object (PDO), and the object dictionary (OD) are reused in the application layer. SDO is used for exchanged parameters and event-trigger data. PDO is used for mapping and un-mapping data frames in real-time communication. The data from the application layer is processed and synchronized in the data link layer by the Field-bus Memory Management Unit (FMMU) and the SyncManager. The data link layer is very important in EtherCAT. It is implemented in an FPGA or application specific integrated circuit (ASIC). For development, we used ET1100 as the data link layer for EtherCAT, which is the main system used for processing data. An EEPROM is connected to an ASIC chip for storing the interface method between the chip and micro controller unit (TMS320F2812). More information about the CiA402 Drive Profile and EtherCAT can found in [20]-[21]. Details on the ISO layered model are presented in [15], [17], [20].

In the EtherCAT based motion control system, the command that controls the stepper motor is embedded into the EtherCAT data frame. The mailbox polling the data from the motor drive every 1 ms. In this paper, we implemented the embedded EtherCAT telegram including a header, a slave ID, a frame type, a data phase, and a working counter. The switch on/off command, position profile, and velocity profile from the master to the drive used this frame . SDO and PDO are used for set and get parameters in network communication. The embedded EtherCAT telegram is shown in Fig. 2 and Table. 1:

III. ETHERCAT BASED OPEN LOOP MOTOR DRIVE

The data in Table 1 is mapped to the EtherCAT transceiver ET1100 Dual PRAM. After that, EtherCAT uses SDO and PDO to extract it in the slave drive. The motion chip uses this data for configuration registers to produce exact pluses for the controller.

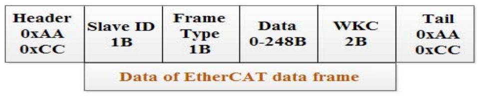

Fig. 2. EtherCAT data frame based motion command

TABLE I. ETHERCAT BASED MOTION COMMAND

Command	Frame type
Set parameters	0x12
Get parameters	0x13
Switch on/off	0x2A
Position Moving	0x34
Velocity Moving	0x37
Homing	0x3D

To develop a step motor controller, modeling is required. This model is very popular [1], [6]-[8] and is described in (1). In order to control the stepping motor, an open-loop or closed-loop controller can be applied. With the open-loop controller, the stepper motor can be controlled in the full step, half step, quarter step, micro-step method, etc.,. The micro-step method provides a good speed performance, reduced resonance, and high torque. Therefore, it is implemented in this motor drive. However, the closed loop control can overcome the lost step and synchronization problems of the stepper motor, which will be presented in Section IV.

$$
\begin{aligned}
\dot{\theta} &= \omega \\
\dot{\omega} &= \frac{1}{J}[-K_m i_a \sin(N_r\theta) + K_m i_b \cos(N_r\theta) - B\omega - T_L] \\
\dot{i_a} &= \frac{1}{L}[v_a - Ri_a + K_m\omega \sin(N_r\theta) \\
\dot{i_b} &= \frac{1}{L}[v_b - Ri_b - K_m\omega \cos(N_r\theta)
\end{aligned} \tag{1}
$$

Here v_a,v_b are the voltages and the currents in phase A and phase B. θ and ω are the rotor position and angular velocity. R and L are the phase resistance and inductance of the motor, respectively. J is the inertia of the motor. N_r is the number of rotor poles. T_L is the load torque of the motor.

A. *Open loop stepping motor*

An overview of the open loop stepper motor is shown in Fig.3. The Pulse Counter receives the command pulse from the motion chip to produce the index for the sine/cosine look up table. In order to rotate the stepper motor, the phase current is controlled following the reference sine/cosine current . The error between the reference current and the feedback current is put into PI controller to produce a PWM duty cycle for switching on/off the H-bridge’s MOSFET. A low pass filter is required for filtering the noise from the feedback current. After that, the signal is amplified to fit the ADC maximum voltage input. More details on the open loop step motor are provided in [2]. A block diagram of the open loop stepper motor is presented in Fig. 3.

However, if torque disturbance occurs, a the high velocity is required, the open-loop can not assure the stepper’s performance. In this case, a closed loop controller for the step motor is necessary. With this method, a lead angle estimation is required.

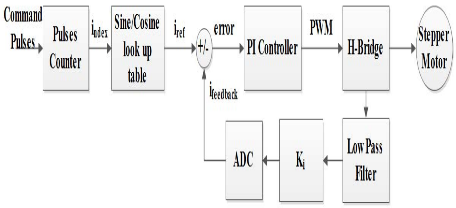

Fig. 3. The open loop stepper motor

B. The lead angle estimation

Because the step motor is operated based on the electro-mechanical mechanism and error between the electrical signal input and rotor position always exists. In order to overcome this problem, a lead angle is required. Normally, a full-step (90^o electrical angle) is added to the position feedback value to correct this error when it operates in the closed-loop method. To achieve a higher velocity, the lead angle value is computed again [2]. In this method, a lead angle estimation based on the speed, EMF, and torque is proposed [4]-[6].

To simplify the lead angle estimation the dq-frame is used instead of (1). The transformation is presented in [5] and is described again in (2):

$$\begin{bmatrix} v_d \\ v_q \end{bmatrix} = \begin{bmatrix} R & -\dot{\theta}L \\ \dot{\theta}L & R \end{bmatrix} + \phi_m \dot{\theta} \begin{bmatrix} 0 \\ 1 \end{bmatrix} \tag{2}$$

$$\begin{cases} v_d = V\cos\gamma \\ v_q = V\sin\gamma \end{cases} \tag{3}$$

$$V = \sqrt{v_a^2 + v_b^2} \tag{4}$$

Where, γ is the lead angle, $L_d = L_q = L$ are assumed constant, ϕ_m is the motor flux, and Z is the impedance of the R and L serial circuit. From (2), the i_q is calculated as:

$$i_q = \frac{V}{Z}\sin(\gamma - \theta) - \frac{R}{Z^2}\dot{\theta}\phi_m \tag{5}$$

If the load torque of the stepper motor is proportional to i_q by K_t times, the lead angle can be regarded as (6):

$$\gamma = \arctan(\frac{\dot{\theta}L}{R}) + \arcsin(\frac{Zi_q}{V} + \frac{R}{ZV}\dot{\theta}\phi_m) \tag{6}$$

IV. ETHERCAT BASED CLOSED-LOOP MOTOR DRIVE

In this section, the closed loop of the stepper motor based on EtherCAT is fully explained. The section is divided into two parts. In Part 1, the data from the EtherCAT frame is extracted. The command pulse generator is presented. In Part 2, the closed-loop stepper motor is introduced.

A. The command pulse generator

The EtherCAT data is processed by the EtherCAT transceiver hardware ET1100 [18] . After that, it continues to be processed by the micro controller TMS320F2812 in the application layer. SPI is used as the interface of MS320F2812 and ET1100 which can support a high baud-rate and has easier implementation than a parallel interface. In order to reduce the memory size and calculation time in the micro controller, a motion chip, PCL6113, is selected to produce a motion profile and pulse calculation for the closed loop stepper motor in Part 2 [19]. The command pulse generator is shown in Fig. 4:

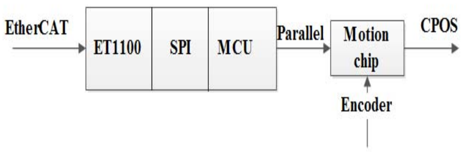

Fig. 4. The command pulse generator.

B. Closed loop stepper motor

The pulses that are produced from the command pulse generator are the closed loop controller input. Fig.5 shows a block diagram of a closed loop stepper motor. The diagram shows the lead angle estimation, the position controller, the current controller. The lead angle estimation is presented in Section III. The position controller made a position profile from 0 to 2 discrete radians during each sampling period. The sample time was 10Khz in the position controller. Here, one counter is set to count and accumulate the pulse that is received from the command pulse generator. Another counter is also set for counting the feed back position value from the encoder.

When the deviation between the reference position and feedback position values is smaller than a 90^o electrical angle, the excitation angle is the reference position. The controller operates in the open-loop mode.

When the deviation between the reference position and feedback position values is larger than a 90^o electrical angle, the controller operates in the closed-loop mode. In order to overcome the lost step, the lead angle is compensated by the value of the feedback position. In some studies [5]-[6], the lead angle selected is a 90^o electrical angle. However, when the motor operates at a vary speed, the torque is changed, EMF appears, and the lead angle at the full step does not assure the performance of the motor. The lead angle in (6) is proposed for position compensation. In this method, the excitation angle is calculated following the calculation of the feedback position and lead angle values. This algorithm is presented in Fig. 6.

The current controller receives the excitation angle as input. The reference current is produced by using the sine/cosine look up table presented in Section III. The formula for the reference current is shown in (7):

$$\begin{aligned} i_a &= I_a\sin(N_r\theta_i) \\ i_b &= I_b\cos(N_r\theta_i) \end{aligned} \tag{7}$$

When the stepper motor operates at a high speed, an EMF occurs. The EMF voltage is a non-linear function. The high

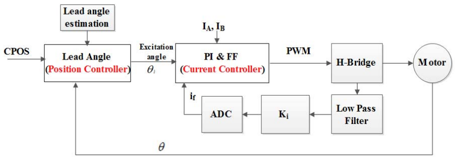

Fig. 5. The closed loop stepper motor.

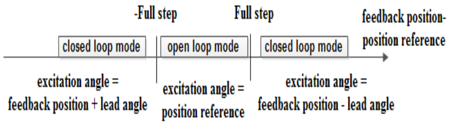

Fig. 6. Position controller algorithm.

current error occurs when just the PI controller is applied as presented in Section III . When the motor operates in the reverse direction, the voltage in the stators needs to be supplied quickly. In order to overcome this problem, a PI-FF controller is proposed. When the FF controller is added to the PI controller, the current error is reduced significantly. A block diagram of the PI-FF controller is shown in Fig. 7:

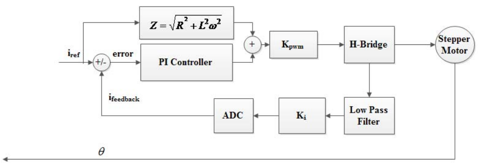

Fig. 7. The PI-FF controller.

V. Experiment Results

To verify the closed loop for the stepper motor using EtherCAT, some experimental results are presented. The implemented system is shown in Fig. 8. A system includes three motor drive connected by EtherCAT network through line topology. Table 2 lists the motor parameters in this experiment. The source voltage for this drive is 24V, the current is 0.84A, the phase inductance is 7.2mH, and the phase resistance is 3.6Ω. In Fig. 9, a cycle time is measured by oscilloscope to

TABLE II. The Motor Parameters

Number of pole	50
Source voltage	24V
Current	0.84A
Phase inductance	7.2mH
Phase resistance	3.6Ω

confirm the real-time system performance through EtherCAT. The cycle time for a system with three drives is 7.92μs. It is very small to provide distributed system.

The reference position and the feedback position are shown in Fig.10. The reference position is plotted with a blue line and the feed back position is plotted with a red line. The results

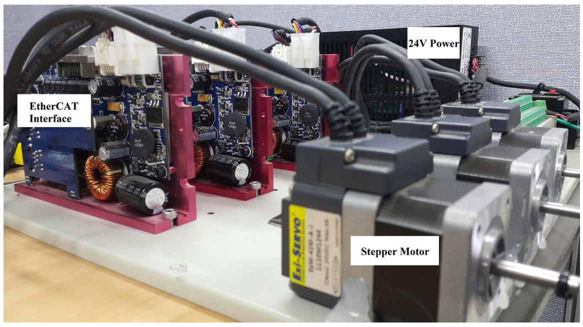

Fig. 8. The experiment system.

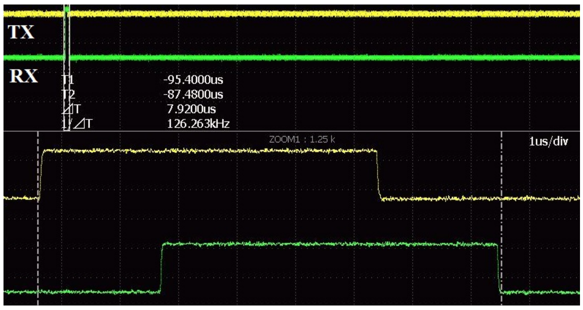

Fig. 9. The EtherCAT cycle time of three axes.

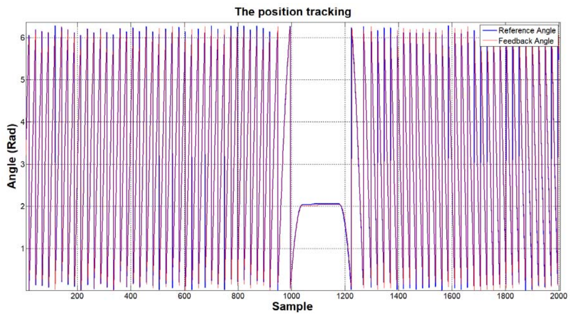

Fig. 10. The position tracking.

indicate that the motor drive has a good performance when the lead angle estimation is used.

A velocity profile is shown in Fig. 11. The feedback velocity is plotted with a red line and the reference velocity is plotted with a blue line. The result confirms the feedback velocity tracking the reference position is very good. A small error is achieved.

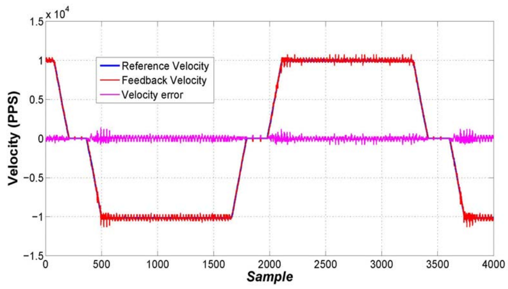

Fig. 11. The velocity tracking.

Fig. 12 presents the PI current controller performance. When the motor runs in the reverse direction, a large error occurs. When applied the PI-FF controller the current can over this problem. The results of PI-FF controller is shown in Fig. 13.

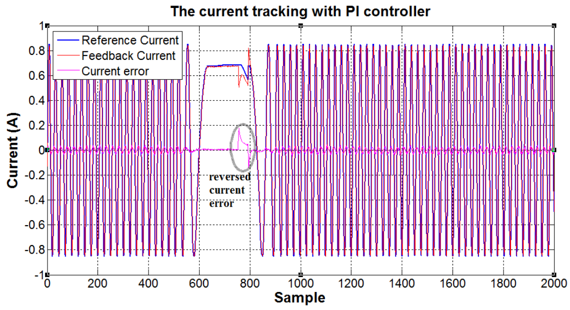

Fig. 12. The current tracking with PI controller.

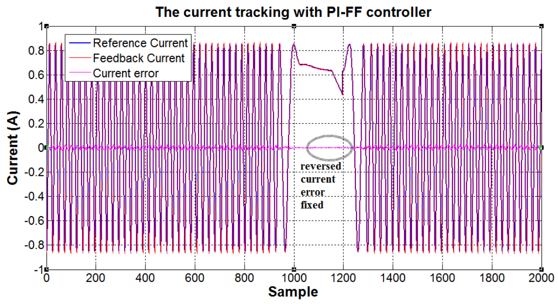

Fig. 13. The current tracking with PI-FF controller.

VI. CONCLUSION

In this paper, we presented a closed-loop controller for a stepper motor using real-time network. In order to overcome the problems with bandwidth, cycle time, and time synchronization, EtherCAT was used. Details on the embedded EtherCAT telegram and CiA402 motion profile were presented. The EtherCAT based open-loop control stepper motor was discussed. An EtherCAT based closed-loop control stepper motor was implemented. A closed-loop position controller that added the lead angle estimation was proposed to achieve smooth velocity and reduce lost steps for the stepper motor. In addition, a PI-FF controller was implemented. This controller is better than a PI controller when the motor runs in the reverse direction. Some experimental results to verify the proposed method were presented. The position, velocity and current of the motor were tracked well. This method is simple and can be applied to a large number of stepper motors in industry. The problems regarding torque compensation and resonance will be researched in the future.

ACKNOWLEDGMENT

This work was supported in part by the Ministry of Science, ICT, and Future Planning, South Korea, under the G-ITRC support program (IITP-2015-R6812-15-0001) supervised by the Institute for Information and Communications Technology Promotion, and in part by the Basic Science Research Program through the National Research Foundation of Korea, funded by Ministry of Education (NRF-2010-0020210).

REFERENCES

[1] Paul Acarnley, *Stepping Motors: a guide to theory and practice*, 4th ed. The Institution of Engineering and Technology, 2007.

[2] Ngoc Quy Le and Jae Wook Jeon, *An Open-loop Stepper Motor Driver Based on FPGA*, International Conference on Control, Automation and Systems, Korea, 2007.

[3] J. B Grimbleby, *Simple algorithm for closed-loop control of stepping motors*, IEEE Proceding in Electric Power Application, January 1995.

[4] Stuart A. Schweid, John E. McInroy, and Robert M. Lofthusy, *Closed Loop Low-Velocity Regulation of Hybrid Stepping Motors Amidst Torque Disturbances*, IEEE Transactions on Industrial Electronics, June 1995.

[5] Ngoc Quy Le, Jung Uk Cho, and Jae Wook Jeon, *Application of Velocity Profie Generation and Closed-Loop Control in Step Motor Control System*, SICE-ICASE International Joint Conference, Korea, 2006.

[6] Quy Ngoc Le, Jae Wook Jeon, *An Improved Method of Speed Damping for a Stepper Motor with a Smooth Speed Estiamtion*, IEEE Intrenational Conference on Robotics and Biomimetics, Thailand, February, 2009.

[7] Jung Uk Cho, Quy Ngoc Le, and Jae Wook Jeon, *FPGA-Based Multiple-Axis Motion Control Chip*, IEEE Transactions on Industrial Electronics, March 2009.

[8] Quy Ngoc Le and Jae Wook Jeon, *Neural-Network-Based Low-Speed-Damping Controller for Stepper Motor With an FPGA*, IEEE Transactions on Industrial Electronics, September 2010.

[9] Byung Yun Oh, Jung Uk Cho, Ngoc Quy Le, Key Ho Kwon, and Jae Wook Jeon, *FPGA-Based motion Controller Using CAN*, International Conference on Control, Auromation and System, Korea, 2007.

[10] Quang Vinh Nguyen, Jae Wook Jeon, *CAN FD-Busload Analysis using the CANoe*, World Congress on Information Technology Applications and Services, 2015.

[11] Rodrigo Lange, Alexsandro C. Bonatto, Francisco Vasques, Rmulo S. de Oliveira, *Timing Analysis of Hybrid FlexRay, CAN-FD and CAN Vehicular Networks*, 42nd Annual Conference of IEEE Industrial Electronics Society (IECON 2016), 2016.

[12] Max Felser, *Real-Time Ethernet Industry Prospective*, June 2005.

[13] Kun Ji, Won-Jong Kim, *Real-Time Control of Networked Control Systems via Ethernet*, International Journal of Control, Automation, and Systems, December 2005.

[14] EtherCAT Technology Group, *EtherCAT Communication: Communication Principle*.

[15] Jin Ho Kim, Jae Hak Lee, Bok Sun Yeom, and Jae Wook Jeon, *Migration from RS-485 to EtherCAT for Closed Loop Step Motor Drive*, IEEE International Conference on Industrial Technology, Korea, 2014.

[16] Vinh Quang Nguyen, Jae Wook Jeon, *EtherCAT Network Latency Analysis*, IEEE International Conference on Computing, Communication and Automation, India, 2016.

[17] Do Eon Lee, Quang Vinh Nguyen. Tae Won Kim, Jun Young Moon, and Jae Wook Jeon, *Development of Independent EtherCAT Slave Module and application to Closed Loop Step Motor Drive with Multi-Axis*, IEEE International Conference on Computing, Communication and Automation, India, 2016.

[18] Beckhoff, *Hardware Data Sheet : ET 1100-EtherCAT Slave Controller*, 2007.

[19] Nippon Pulse Motor Co., Ltd, *Users Manual for PCL6113/6123/6143 Pulse Control LSI*, 2005.

[20] EtherCAT Technology Group, *EtherCAT Communication Specification*, Version 1.0, 2004.

[21] ETG, *Proposed Enhancements to CiA 402 drive profile to be included in IEC 61800-7*, 2005.