Proceedings of 2018 IEEE
International Conference on Mechatronics and Automation
August 5 - 8, Changchun, China

A Control System Design for 7-DoF Light-weight Robot based on EtherCAT Bus

Guojun Zhang, Fenglei Ni*, Zhiqi Li, Hong Liu
State Key Laboratory of Robotics and System
Harbin Institute of Technology
Harbin, Heilongjiang Province, China
Corresponding author*: flni@hit.edu.cn

***Abstract*— In order to increase control frequency and expansibility for 7-degree-of-freedom(DoF) light-weight, a novel robot control system based on EtherCAT bus is designed. The control system consists of arm controller and joint controller. The arm controller is implemented by a core-i5 PC, which achieves motion control, trajectory planning and real-time task planning. The joint controller is designed based on Field Programmable Gate Array(FPGA), which achieves sensor samping and motor control. EtherCAT bus achieves data transmission between joint controller and arm controller. The experiment results show that our control system can achieve higher control frequency and better expansibility.**

***Index Terms*—EtherCAT bus, Light-weight robot, Control system**

I. Introduction

With the tendency of the industry 4.0, manufactory is becoming more and more flexible and intelligent [1], [2]. Traditional industrial robots have more strength, high speed, robustness, and positional accuracy. Hence, they are always working in the cage without human [3], [4]. And they could not meet the robot-human cooperation in the intelligent factory [5]. Nowadays, more production operations need human and robot work together. It is necessary to get more sensor information to prevent the risk of human harm [6]. So, more sensors need to be integrated in the robot, for instance, torque sensor, encoder, current and voltage sensor. More sensors will cause more information and stress transmission. The 1Mbit/s rate of CAN bus couldn't meet the transmission of more information [7], [8]. The 100Mbit/s rate of Real-time Ethernet protocols, such as PROFINET protocol and EtherCAT protocol, have solved this problem [9].

By contrast, EtherCAT protocol offers significantly better performance than PROFINET protocol in aspects that supplying more slave devices up to 65535, shorter cycle time below 100μs, better performance of synchronization [10]. EtherCAT bus relies on a clock synchronization mechanism matches any term of the IEEE 1588 precision time protocol(PTP), which is known as distributed clock(DC). The DC mechanism is able to deal with aspects such as the initial offset and propagation delay by compensating statically [11], [12]. Due to this, EtherCAT bus achieves accurate synchronization, which clock deviation is well below 1 us between master and slaves.

A typical robotic system based on ETherCAT bus contains a master(a PC with real-time operation system) and slaves(joints of the robot) [13]. Of course, to take full advantage of EtherCAT bus, the excellent master and slave are needed. There have been many EtherCAT masters that relied on different platforms and many designs of slaves with different Microcontroller Units(MCU) [14], [15]. In [16], [17], they analyzed open-source real-time extensions for Linux such as RT Linux, RTAI, Xenomai, and AQuoSA. According to their works, Xenomai has better real-time accurate. In [18], Zhou Yang et al. proposed a Decentralized control system based on a PCI-based DSP. It just achieved 1ms cycle time. In [19], LI Chun-Mu et al. designed an EtherCAT slave module based on DSP, which used the Serial Peripheral Interface(SPI) between DSP and EtherCAT slave controller(ESC), it dosen't achieve high transmission rate.

In this paper, we propose a new centralized control system for 7-DoF light-weight robot, which is integrated trajectory planning, control algorithm, real-time task planning into the arm controller. Moreover, we designe a new joint controller based on the FPGA, which has some advantages contain the high sampling frequency, the ability to re-program in the hardware [20], [21], we choose Altera FPGA as the MCU of the controller of robotic joint, and ET1100 as the ESC. To get shorter system control period, the update rate of sensors must be faster. Using Very-High-Speed Integrated Circuit Hardware Description Language(VHDL) could get fast enough update of all sensors [22], [23].

This paper will be organized as follows: A real-time control system of 7-DoF robot based on EtherCAT bus is described in Section II, including the design of master and design of joint. In Section III, the centralized control of the robot is described. Finally, the experiment results are shown in Section IV.

II. The control system design based on EtherCAT bus

To improve control performance for 7-DoF light-weight robot, we build a control system based on EtherCAT bus. This control system is composed of arm controller and joint controller. Thereinto, the arm controller achieves motion control and trajectory planning. The joint controller achieves sensor sampling and motor control.

978-1-5386-6075-1/18/$31.00 ©2018 IEEE

A. EtherCAT bus

EtherCAT bus is a kind of real-time industry Ethernet technology, which takes full advantage of full duplex characteristic of Ethernet. A typical EtherCAT system contains a master and many slaves. Master sends and receives a standard Ethernet frame. These slaves can handle directly received datagram, and extract or insert related user data from datagram. Then the dealt datagram is transmitted to next EtherCAT slave. The last slave returns complete processed datagram to master.

The performance of robotic motion control depends on servo-control. EtherCAT servo driver supports the application of CANopen protocol and SERCOS protocol, which named CoE and SOE. In our work, we choose CoE in our robotic servo-control system.

The architecture of CoE device is shown in Fig. 1. A control application registers all objects that map PDO message. Every PDO entry is accessed synchronously by a sync manager of ESC. Service data objects(SDO) are accessed by mailbox when some nonperiodic service needs to be responded.

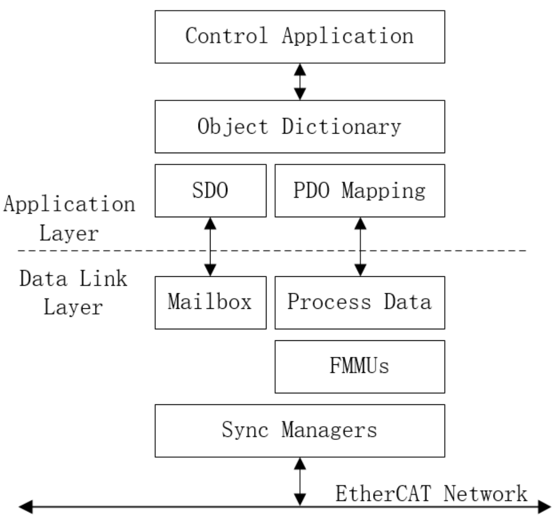

Fig. 1 The Architecture of CoE device

EtherCAT master devices have two synchronous modes including cyclic mode and DC mode. The DC mode has more accurate than the cyclic mode. Hence, the servo control of robotic must run on the DC mode. On the DC mode, the cyclic period of EtherCAT slave device is triggered by SYNC event. As shown in Fig. 2. All slave devices should be synchronized by the clock of DC system. As the same with slave, Master should be synchronized by the clock that references the clock of first slave device. The mechanism of data transmission of DC system is shown in Fig. 3.

B. The arm controller

In this paper, the light-weight robot consists of 7 serial joints, which is shown in Fig. 4. In this robot arm, one joint controller board including two motor drive modules can control two adjacent motors. So, there are 4 EtherCAT

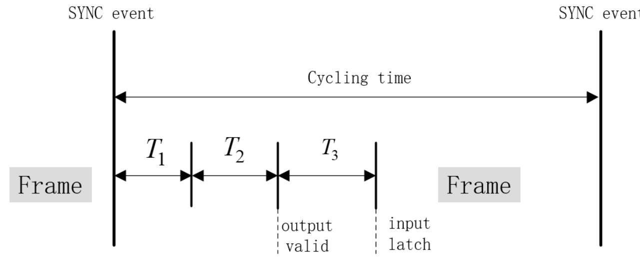

Fig. 2 Local cycles synchronized to the Sync event

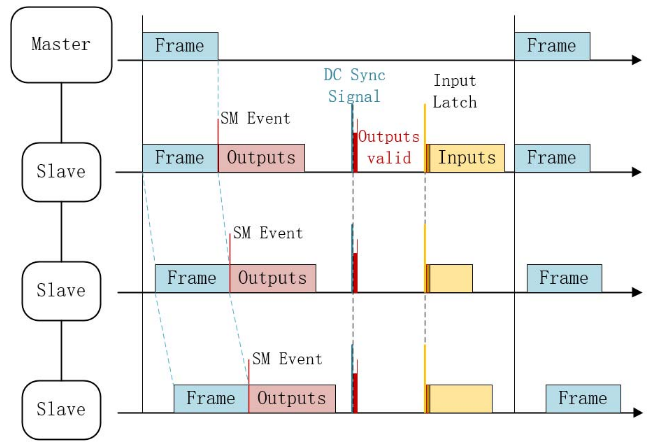

Fig. 3 The mechanism of DC system

slaves connected by linear topology. Due to characteristics of EtherCAT bus, a hard real-time system is required. Especially for a light-weight robot that is highly dynamic and accurate to work in unstructured environment, the high real-time performance of system is strongly recommended.

According to above requirements, a high performance PC was chosen as master, which consists of Intel Core i5-4460 3.4GHz, 4GB DDR3 RAM, and 120GB HDD. As shown in Fig. 5, an entire EtherCAT master consists of user space, real-time kernel and EtherCAT master stack. In the user space, the user can plan the Non-RT tasks and RT tasks. The master operates on the Xenomai 3.0.2 based on the Linux 3.18.20. The RT tasks must be operated on the Xenomai real-time kernel that connects the EtherCAT master stack by the relevant APIs. There is the EC-Master on the real-time OS, which is produced by the Accotis Company. In the EtherCAT Master Core cyclic process data and acyclic mailbox commands are sent and received. The EC-Master is configured by a XML file whose format is defined by the EtherCAT specification ETG.2100. EC-Master includes an OS independent XML parser. This XML file records the EtherCAT bus topology and the frames to exchange with the slave devices. At the EC link layer, the Ethernet frames are handled between the master and the slave devices. When the system need the hard real-time requirements, this layer has to be optimized for the network adapter card that is compatible with EtherCAT protocol. Then, all OS dependent system calls are packaged in the OS layer.

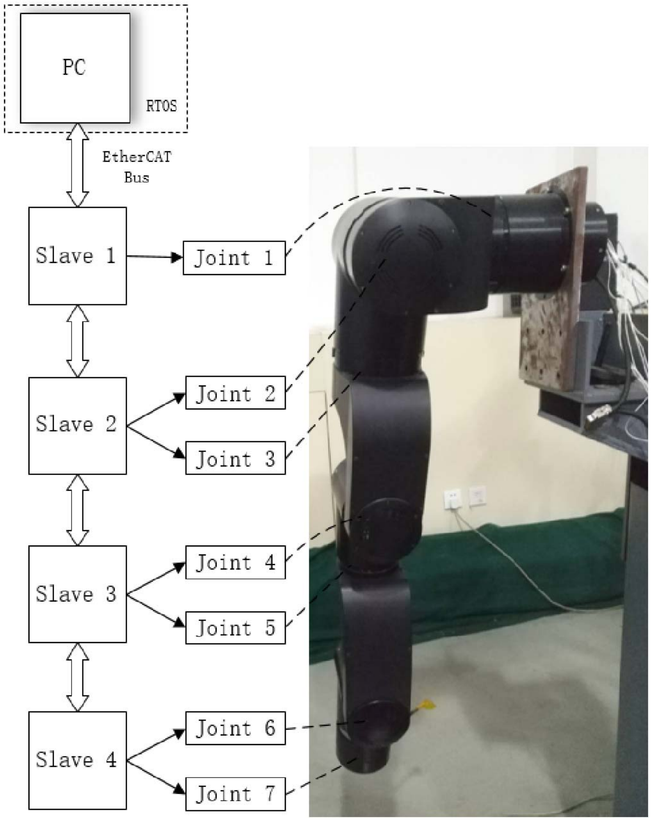

Fig. 4 The system of 7-DoF light-weight robot

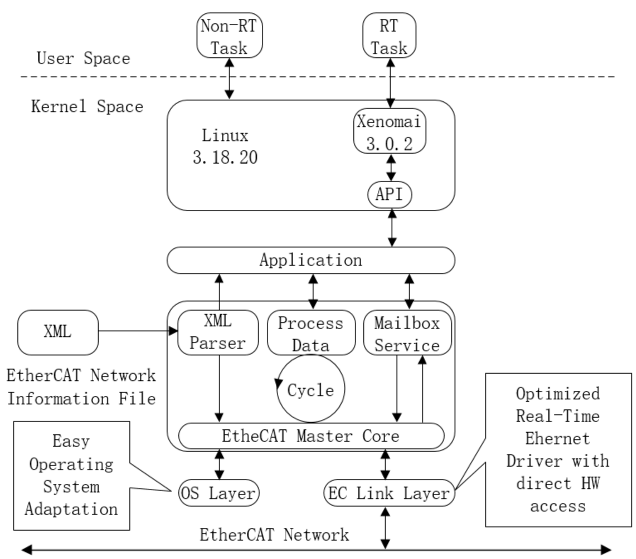

Fig. 5 The Architecture of the PC

C. The joint controller

In this paper, the motor driver and controller board are embedded into the structure of joint. Every joint is serially connected with the master computer via EtherCAT bus. Due to the embedded design, only six wires inside the robot are needed to connect all joints with the outer controller PC and the power supply. Modular design is able to minimize the difference of structure of joints. The structure of joint is shown in Fig. 6. Modular joint consists of motor, Harmonic Drive gears, two encoders, torque sensor of joint, bearings, safety brake, motor driver, controller board of joint and power supply. Additionally, one of two encoders is used to measure the position of motor, another is used to measure the position of joint. The torque sensor is assembled between the flex spline of the Harmonic Drive gear and the link of joint, which measures the joint torque acting on the link.

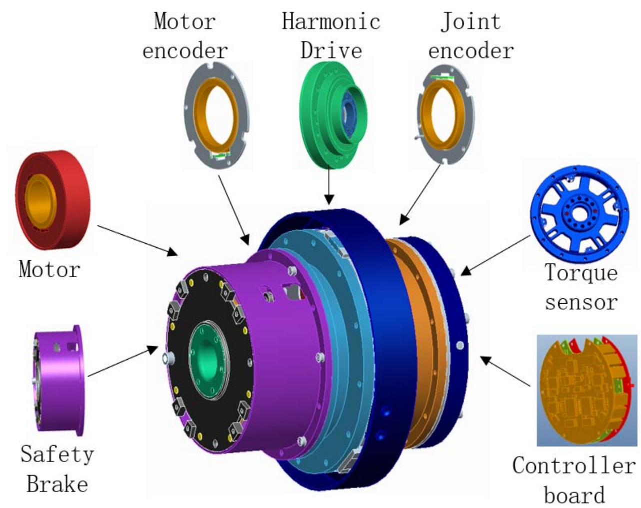

Fig. 6 The mechatronic design of light-weight robot's joint

The joint controller board was designed based on FPGA. According to the EtherCAT, the hardware of the joint controller has to contain those aspects as fellows: application layer, data link layer and physical layer. The physical layer includes standard Ethernet PHY chips, isolation transformers, standard RJ45 wire connectors. Moreover, in data link layer, ET1100 transmits and parses EtherCAT frames between physical layer and application layer. The FPGA can control servo motors, sample sensors data with high speed, extract and insert user data from EtherCAT frame. ET1100 supports several kinds of PDI, which can be configured as digital I/O interfaces, SPI, asynchronous 8/16 bit parallel interface and synchronous 8/16 bit parallel interface. In order to achieve high frequency motor control, the asynchronous 16 bit parallel interface is used as PDI between FPGA and ESC. Fig. 7 shows the connection diagram of asynchronous 16 bit parallel interface. The ET1100 is configured by EEPROM when the power supplied every time. Similarly, FPGA is configured by flash device where stored the program of joint controller. With the advantages of FPGA, the joint controller can integrate all kinds of functions, e.g: sampling sensors, communication and motor driving. Therefore, many high speed sensor interfaces were programmed by VHDL language on the FPGA as shown in Fig. 8. We built some intellectual property(IP) cores to sample data from sensors over 1MHz, which is fast enough for system feedback control. As follows, torque interface is the IP core to read data from the chip of torque sensor. The field-oriented control(FOC) module can control permanent magnet synchronous motor(PMSM) efficiently, which includes encoder

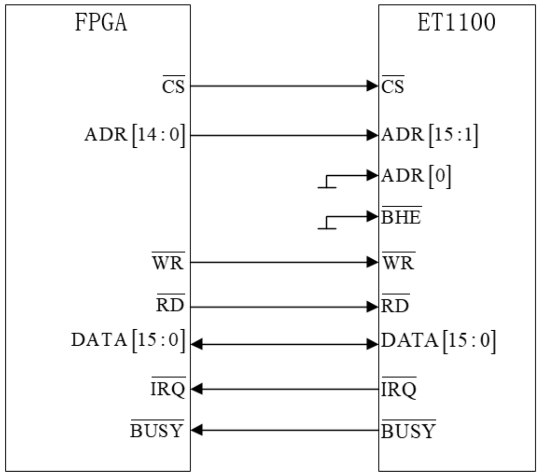

Fig. 7 Connection with 16 bit μControllers

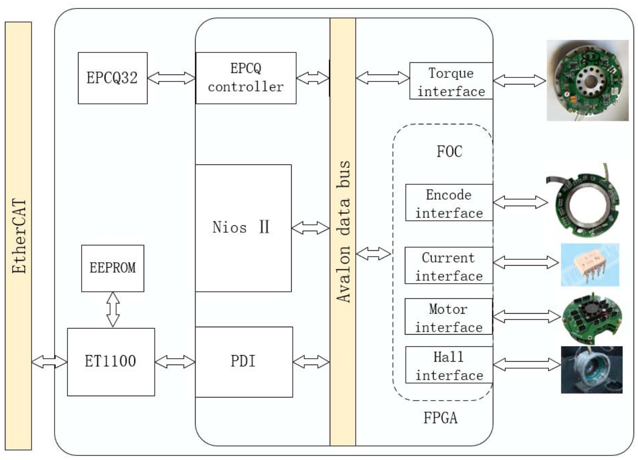

Fig. 8 System diagram of joint controller based on FPGA

interface, current sensor interface, motor driver interface and hall sensor interface.

III. The control strategy for light-weight robot

This section will mainly focus on the specific control method based on the new hardware and the operating conditions with humans in unstructured environments. Higher control frequency can lead better motion accuracy and promote system response speed. So, we use above hardwares to design a new faster robot control system.

A. The control strategy of 7dof light-weight robot

Almost proposed joint space control techniques could be divided into two kinds of control method which are centralized control scheme and decentralized control scheme. The former is used in controlling single manipulator joint independently of the other joints. Another, centralized control scheme is applied when considering the dynamic mutual effects between joints. The decentralized control scheme regards the manipulator that formed by n independent systems and control every joint axis as a simple single-input/single-output system. The coupling effects between joints caused by varying configurations during motion are regarded as disturbance inputs. Nonetheless, when great operational speeds are

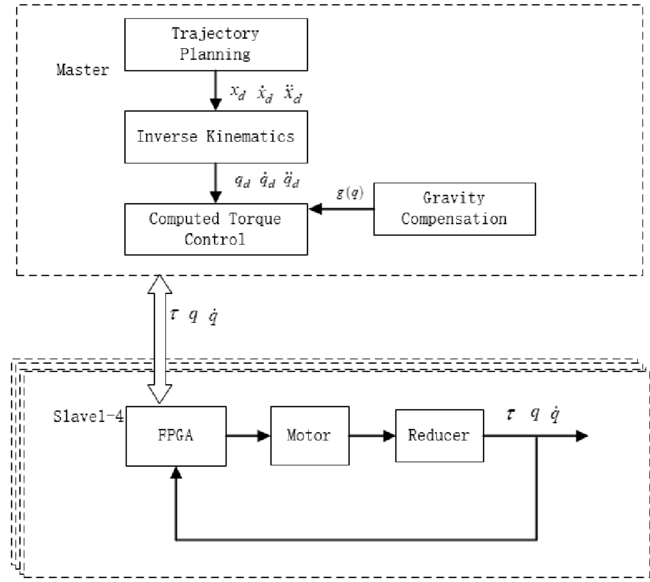

Fig. 9 Structure of centralized control of light-weight manipulator

needed, the nonlinear coupling terms strongly affect system performance. In this condition, it is essential to exclude the causes by generating compensation torques for the nonlinear terms. This is centralized control algorithms based on the manipulator dynamic model.

In this paper, a novel hardware control structure is proposed as shown in Fig. 9. The PC is used to compute trajectory planning, forward/inverse kinematics, gravity compensation and control algorithm. Using PC could reduce the floating points computation time and promote the control frequency.

B. Joint level control

At joint level, a nonlinear state feedback controller is applied by using the entire joint state in the control feedback loop. The dynamic math model of an n-joint robot arm is expressed as

$$B(q)\ddot{q} + n(q,\dot{q}) = \tau \tag{1}$$

Where, has been set

$$n(q,\dot{q}) = C(q,\dot{q})\dot{q} + F\dot{q} + g(q) \tag{2}$$

In fact, the equation in (1) is linear with the control τ, and has a full-rank matrix $B(q)$ which could be inverted by any robot arm configuration. Regarding the control τ as a function of the robot arm state in the form

$$\tau = B(q)y + n(q,\dot{q}) \tag{3}$$

results in the system expressed by $\ddot{q} = y$ According to the equation (3), the manipulator motion control problem is felled into that of finding a stable control law y. For this purpose, the choice

$$y = -K_P q - K_D \dot{q} + r \tag{4}$$

results in the system of second-order equations

$$\ddot{q} + K_D \dot{q} + K_P q = r \tag{5}$$

Where, assuming the K_P and K_D are positive definite matrices. The equation (5) is asymptotically stable. Setting K_P and K_D as diagonal matrices of the type $K_P = diag\{\omega_{n1}^2, ..., \omega_{nn}^2\}$ $K_D = diag\{2\zeta_1\,\omega_{n1}, ..., 2\zeta\omega_{nn}\}$ obtains a decoupled system. The joint variable q_i is only influenced by the reference component r_i, which both are a second-order input/output relationship that depended on a natural frequency ω_{ni} and a damping ratio ζ_i. Given arbitrary desired trajectory $q_d(t)$, tracking of desired trajectory for the output $q(t)$is assured by choosing

$$r = \ddot{q}_d + K_D\dot{q}_d + K_P q_d \tag{6}$$

substituting (6) into (4) obtaining the control law y

$$y = \ddot{q}_d + K_D\dot{e}_q + K_P e_q \tag{7}$$

substituting (7) into (1), the overall control input becomes

$$\tau = B(q)(\ddot{q}_d + K_D\dot{e}_q + K_P e_q) + C(q,\dot{q})\dot{q} + F\dot{q} + g(q) \tag{8}$$

In practice, substituting (6) into (5) resulting linear error dynamics are

$$\ddot{e}_q + K_D\dot{e}_q + K_P e_q = 0 \tag{9}$$

As illustrated in Fig. 10, which contains two feedback loops: an inner loop based on the robot arm dynamic model, and an outer loop handling with the tracking error. The purpose of the inner loop is to get a decoupled and linear input/output relationship. Furthermore, the outer loop is required to stabilize the entire system.

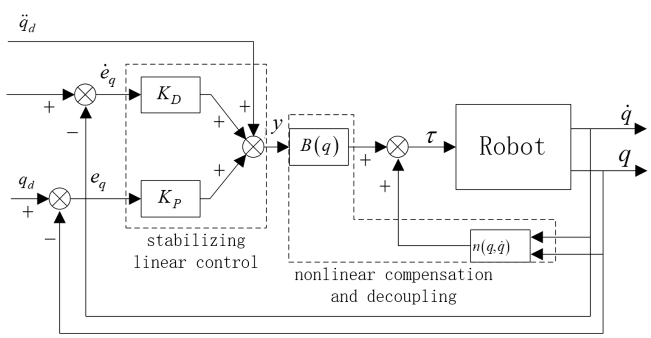

Fig. 10 Schematic diagram of joint space inverse dynamics control

IV. Experiment

The greater bandwith of system can be conducive to promote system control accuracy. However, too shorter cyclic control time will cause a mass of computation. So, we used a powerful computer as the master of light-weight control system. control method equation(8) applied in single joint control to follow the desired trajectory of three-order curve as shown in Fig. 11. Set parameter K_D=1000, K_P=10. The results of three control experiments, which cycle time is 5ms , 1ms, 200μs. The Fig. 12 shows the position tracking errors curve changes over time of those three experiments. We can see from Fig. 12:

(1) With 5ms cyclic time, the static error of joint is 0.021°and dynamic error is 0.03°.

(2) With 1ms cyclic time, the static error of joint is 0.01°and dynamic error is 0.02°.

(3) With 200μs cyclic time, the static error of joint is 0.007°and dynamic error is 0.015°.

In a result, with enhancing the speed of communication bus, the performance of joints tracking is promoted significantly.

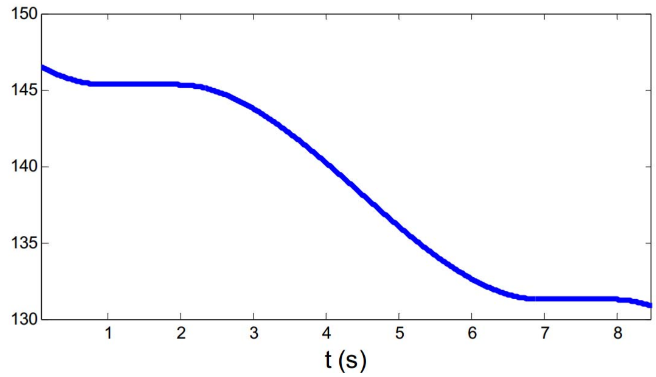

Fig. 11 The desired trajectory of three-order curve

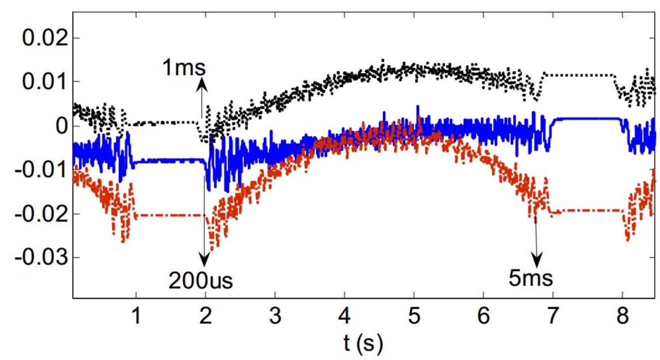

Fig. 12 The position tracking error curve changes over time

V. Conclusion and Future Work

In this study, we built a real-time robot control system based on EtherCAT bus. The EtherCAT master was developed under Linux environment with Xenomai kernel. And the slaves were designed based on FPGA, which control the motion of joints. The FPGA provides the advantages of high speed and parallel data processing to joint controller. Through experiments, the new centralized control system can achieve higher control frequency(up to 20KHz). And, the robot motion control accuracy can be promoted in higher control frequency, which is clearly shown from the above experimental results.

In the future, we will continue to increase the motion control performance for the light-weight robot.

Acknowledgement

This study is supported by the the Foundation for Innovation Research Groups of the National Natural Science Foundation of China (Grant No.51521003).

REFERENCES

[1] S. Wang, J. Wan, D. Zhang, D. Li, and C. Zhang, "Towards smart factory for industry 4.0," *Computer Networks the International Journal of Computer and Telecommunications Networking*, vol. 101, no. C, pp. 158–168, 2016.

[2] J. Lee, B. Bagheri, and H. A. Kao, "A cyber-physical systems architecture for industry 4.0-based manufacturing systems," *Manufacturing Letters*, vol. 3, pp. 18–23, 2015.

[3] D. E. Whitney, "Industrial robot forward calibration method and results," *Journal of Dynamic Systems Measurement and Control*, vol. 108, no. 1, pp. 1–8, 1986.

[4] J. Swevers, W. Verdonck, and J. De Schutter, "Dynamic model identification for industrial robots," *Control Systems IEEE*, vol. 27, no. 5, pp. 58–71, 2009.

[5] MASATAKAYOSHIMURA, NAOHIROYOSHIKAWA, and E. Katsiwdohitomip., "Design optimization of industrial robots considering the working environment," *International Journal of Production Research*, vol. 28, no. 5, pp. 805–820, 1990.

[6] A. AlbuSchffer, S. Haddadin, C. Ott, A. Stemmer, T. Wimbck, and G. Hirzinger, "The dlr lightweight robot: design and control concepts for robots in human environments," *Industrial Robot*, vol. 34, no. 5, pp. 376–385, 2007.

[7] M. Van Osch and S. A. Smolka, "Finite-state analysis of the can bus protocol," in *IEEE International Symposium on High Assurance Systems Engineering*, 2001, pp. 42–52.

[8] M. Gergeleit and H. Streich, "Implementing a distributed high-resolution real-time clock using the can-bus," 1994.

[9] J. D. Decotignie, "The many faces of industrial ethernet [past and present]," *IEEE Industrial Electronics Magazine*, vol. 3, no. 1, pp. 8–19, 2009.

[10] G. Prytz, "A performance analysis of ethercat and profinet irt," in *IEEE International Conference on Emerging Technologies and Factory Automation*, 2008, pp. 408–415.

[11] G. Cena, I. C. Bertolotti, S. Scanzio, A. Valenzano, and C. Zunino, "Evaluation of ethercat distributed clock performance," *IEEE Transactions on Industrial Informatics*, vol. 8, no. 1, pp. 20–29, 2012.

[12] G. Cena, I. C. Bertolotti, S. Scanzio, and A. Valenzano, "On the accuracy of the distributed clock mechanism in ethercat," in *IEEE International Workshop on Factory Communication Systems*, 2010, pp. 43–52.

[13] Z. M. Liu, N. L. Liu, T. Zhang, L. Cui, and H. Y. Li, "Ethercat based robot modular joint controller," pp. 1708–1713, 2015.

[14] C. Kang, Y. Pang, C. Ma, and C. Li, "Design of ethercat slave module," in *International Conference on Mechatronics and Automation*, 2011, pp. 1600–1604.

[15] B. Ma, W. Yao, Y. Liu, X. Zhang, B. Huang, and D. Zhao, "Analysis and verification of ethercat slave controller communication link based on fpga," *Application of Electronic Technique*, 2017.

[16] Y. Moon, N. Y. Ko, K. Lee, Y. Bae, and J. K. Park, "Real-time ethercat master implementation on xenomai for a robot system," *International Journal of Fuzzy Logic and Intelligent Systems*, vol. 9, no. 3, pp. 244–248, 2009.

[17] M. Cereia, I. C. Bertolotti, and S. Scanzio, "Performance of a real-time ethercat master under linux," *IEEE Transactions on Industrial Informatics*, vol. 7, no. 4, pp. 679–687, 2011.

[18] Y. Zhou, B. Wang, L. Jiang, and H. Liu, "Rapid prototyping of real-time controllers for humanoid robotics: A case study," in *IEEE International Conference on Robotics and Biomimetics*, 2012, pp. 2339–2344.

[19] L. I. Chun-Mu, M. Z. Zhang, X. W. Zheng, L. I. Yin, and Y. M. Chen, "Application of ethercat fieldbus in servo motion control system," *Journal of Mechanical and Electrical Engineering*, 2011.

[20] J. Guajardo, S. S. Kumar, G. J. Schrijen, and P. Tuyls, "Fpga intrinsic pufs and their use for ip protection," in *International Workshop on Cryptographic Hardware and Embedded Systems*, 2007, pp. 63–80.

[21] E. Monmasson and M. N. Cirstea, "Fpga design methodology for industrial control systemsa review," *IEEE Transactions on Industrial Electronics*, vol. 54, no. 4, pp. 1824–1842, 2007.

[22] K. Skahill, J. Legenhausen, R. Wade, C. Wilner, and B. Wilson, "Vhdl for programmable logic," 1996.

[23] P. J. Ashenden, G. D. Peterson, and D. A. Teegarden, *The designer's guide to VHDL /*. Morgan Kaufmann,, 2002.