REAL TIME PC BASED CONTROL OF PARALLEL ROBOT

[1]SUN LIM, [2]HAK SANG JUNG, [3]IL-KYUN JUNG

[1,2,3]Intelligent Robotics Research Center, Korea Electronics Technology Institute(KETI), Bucheon-si, South of Korea
E-mail: sunishot@keti.re.kr, haha0000@keti.re.kr, mickey3d@keti.re.kr

Abstract- In this paper, we describe the control strategy of high speed parallel robot system with EtherCAT network. This work deals the parallel robot system with centralized control on the real-time operating system such as window RTX, intime etc. Most control scheme and algorithm is implemented master platform on the PC, the input and output interface is ported on the slave side. The data is transferred by maximum 20usecond with 1000byte. EtherCAT is very high speed and stable industrial network. The control strategy with EtherCAT is very useful and robust on Ethernet network environment. Our system is selected window based Twincat EtehrCAT master platform for users. The system has very advantage for implementation and running of the control system.

Keywords- Parallel Robot Control, Ethercat, Realtime-Control

I. INTRODUCTION

The Parallel Robot as Shown in Fig 1(a)(b) is one of the successful industrial robot in many application domains, pick-and-place, packing even assembling. Because of this, Parallel Robot have many advantages compared to traditional serial robots, such as speed and accuracy. based on the end-effector of parallel robot can move a light weight objects at a speed of several meter per second, and has high accuracy within a limited range as shown in Fig 1(c).



Figure 1. Parallel Robots

In addition, a mechanism that operates in a limited range to ensure safety in exceptional circumstances due to a device that acts as an element of performance and safety were met. In recent years, several devices in end-effector by installing a complex assembly and operation is possible as shown in Fig 2. However, demand in industry turned to small quantity batch production, As the conversion process can be done easily and quickly rose in the interest of the system. At the traditional robot environment, the changes in the process of time it took several months. In addition, if the accuracy requirements of the job took a long time used to verify.

In this paper, we present a centralized parallel robot control system for rapid process changes. The proposed system changes manufacturing process through the software and a number of parallel robots have the flexibility to expand as high-speed synchronization via industrial networks by supporting to improve the performance of a parallel robot control.

This paper is structured as follow: In Section 2, the background technologies of presented system are described. In Section 3, Control strategy and concept of the system are described. Finally, Section 4 describes the concluding remark.

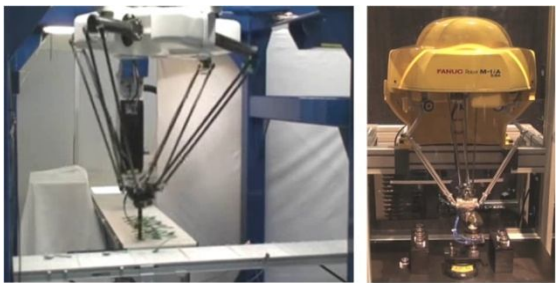


Figure 2. Parallel Robots in Industrial Field. Pick-and-Place(left) and Assembling (right)

II. BACK GROUND

A. Real-Time Operating System

Most of the robot control systems are being developed using window operating system. Windows has many advantage of easy developing environment, such as, supporting abundant device drivers and multitasking. However it cannot support hard real time capabilities since the windows is not designed for the real-time system originally to emphasize the graphic display and to use the multi task in viewed same time. Moreover it is difficult and takes a lot of time to develop the device drivers in windows operating system. Basically a real time O/S is suitable to implement a real time control system.

The commercial QNX and VxWorks have the hard real time capabilities. However, the price is too high to be used for education and research purposes. Their support of device drives for custom designed controllers are so weak that the expendability is very low. Thus our system is ported to window real time operating system , such as, window-RTX. This OS have an advantage of the similar window OS and hard real time is guaranteed.

B. EtherCAT

EtherCAT is Real Time Ethernet technology based on the Ethernet. Its objective is to maximize the utilization of the full duplex Ethernet bandwidth. The network communication system employs the Master/Slave principle. The master node send the packet frames to the slave nodes. The slave node extract data from and insert data in to packet frames. Telegram processing principles are utilized "on the fly". In this way, by performing EtherCAT can use ninety percent of bandwidth without packet collision. Also EtherCAT provide distributed clock function for synchronization with each slave node. It can be improve the control capability of parallel robots

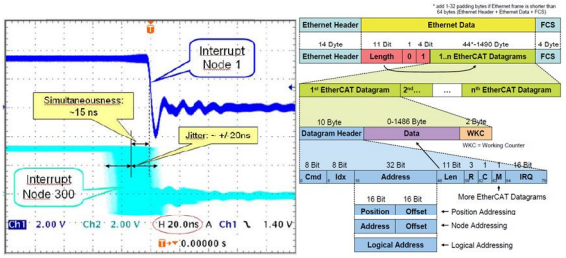


Figure 3. Accuracy Result of Distributed Clock by EtherCAT (left) and EtherCAT Datagram (right)

II. ETHERCAT BASED PARALLEL CONTROL STRATEGY

Figure 4. illustrate the Delta robot that is one of the industrial robots more used today and is the most successful robots which have been designed. The parallel robot is useful its applications include pick, place and packing products in the food, medical and cosmetic industry, as well as in assembly process of electronic device. The parallel robot is made up of an active joint with joint motor and passive joint which are linked together three independent rigid arms. The actuators are placed for each arm as shown in Fig 3. right side.

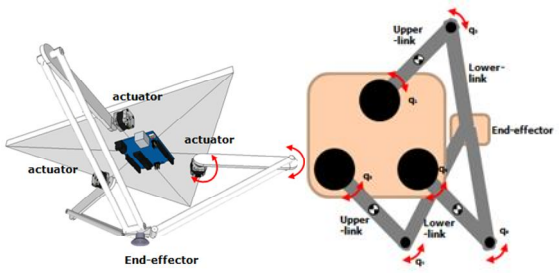


Figure 4. The Parts of parallel robot (left) and configuration (right)

A. Parallel Robot Master Controller

Master Controller is made up of industrial PC platform and EtherCAT Master Stack. The system is implemented window OS such as window 7-32bit, with window RTX. The parallel robot control scheme and safety mechanism is running on window RTX for real-time and periodic computing work. The monitoring and user interface is implemented window side. The data is shared between window and RTX with inner interface socket communication. Figure 7. illustrate the Master Controller structure. This structure is very practical on industrial controller.

The control scheme is detailed in figure 5. The robot control attempts to achieve motion control using parallel robots under unknown manufacturing tolerances and inaccuracies by migrating the measurement from joint space and tack space. This problem is solved by the on and off calibration of the parallel body. However this is very hard and time consuming work so most field engineer want to avoid this work and automatically find the parameter by program. Thus the control objective of this paper is described as bellows;

Find the control law below condition is satisfied as

$$\lim_{t\to\infty}\begin{bmatrix} x_i^* \\ y_i^* \\ z_i^* \\ t_i^* \end{bmatrix} - \begin{bmatrix} x_i \\ y_i \\ z_i \\ t_i \end{bmatrix} = 0 \longrightarrow \begin{bmatrix} e_i^x \\ e_i^y \\ e_i^z \\ e_i^t \end{bmatrix} = 0 \qquad (1)$$

and

$$\tau(t) \le \alpha < \infty$$

where * upper script mean the command and lower script i mean the i-index parallel robot platform. x, y, z, t are a Cartesian space axis, individually x-axis , y-axis, z-axis and time domain. The task space control is very difficult because feedback of the end-effector on the cartesian space is not useful measure on real-time. Thus most control scheme is designed on the joint space of parallel robot. This control block diagram is represented on Figure 4. This work attempts to achieve precise motion control using parallel robot that encounters manufacturing tolerances, assembly errors and thermal deformations without periodic calibration through only joint space measurements.

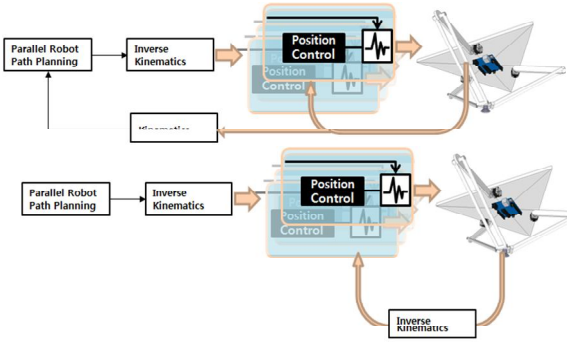


Figure 5. The Joint space control block diagram(upper) and the task space control block diagram(lower).

The parallel robot path planning block is the generation block of motion on the cartesian space trajectory or regulation. This block make the x,y,z and time trajectory that is input of the inverse-kinematics

block. The inverse-kinematic block generate the joint space motor position trajectories.

The parallel robot dynamic equation of motion in the configuration space can be written as follows

$$M(q)\ddot{q} + b(q,\dot{q}) + g(q) = \tau(t) \quad (2)$$

where $M(q)$ is the parallel robot positive definite inertia matrix, $b(q,\dot{q})$ is a vector of coriolis and centripetal forces, $g(q)$ is the gravity term while is the generalized torque vector acting on the generalized coordinate vector. Most practical system have the physical measurement uncertainty so taking the dynamical inaccuracies into consideration, one can write the following relations

$$\begin{aligned} M(q) &= M_n(q) + \Delta M(q) \\ g(q) &= g_n(q) + \Delta g(q), \\ b(q,\dot{q}) &= b_n(q,\dot{q}) + \Delta b(q,\dot{q}) \end{aligned} \quad (3)$$

where $M_n(q), b_n(q,\dot{q}), g_n(q)$ are respectively the nominal value inertia matrix, nominal vector of coriolis and centripetal forces and the nominal gravity term. These value are measured by the calibration tool or 3D cad software. Δ symbol stands for the deviation between these terms and the actual ones.

Rewriting (2)

$$M_n(q) + b_n(q,\dot{q}) + g_n(q) + \Delta M(q) + \Delta b(q,\dot{q}) + \Delta g(q) = \tau(t)$$

$$M_n(q) + b_n(q,\dot{q}) + g_n(q) + d(t) = \tau(t) \quad (4)$$

Where $d(t) = \Delta M(q) + \Delta b(q,\dot{q}) + \Delta g(q)$ is disturbance term that is canceled by the disturbance observer which is described as follows

The dynamic uncertainties such as the un-modeled dynamics are considered as disturbances, then a disturbance observer is designed to estimate such disturbances and used to generate compensation term.

$$\hat{d}(t) = \tau(t) - \tau^*(t) \quad (5)$$

where $\hat{d}(t)$ is disturbance estimation value and $\tau^*(t) = M_n(q) + b_n(q,\dot{q}) + g_n(q)$ is a calculation value with nominal value.

Moreover, it can be estimated through the following low pass filter for avoiding noise sensitivity,

$$\hat{d}(t) = \frac{h}{s+h}\left[\tau(t) - \tau^*(t)\right] \quad (6)$$

where h is the single observer gain which controls how fast the estimated signal converges to the actual disturbance.

Then, the control law is represented as follow

$$\tau(t) = \frac{1}{M_n}\left[b_n(q,\dot{q}) + g_n(q) - ke(t) + \hat{d}(t)\right] \quad (7)$$

where k is the control gain for reducing convergence time and regulation error.

The stability is proved by the lyapunov function that is omitted in this work.

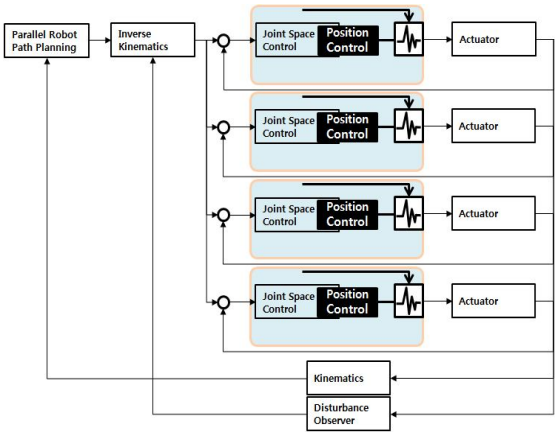


Figure 6. The Parallel robot control block diagram

Overall this control scheme is running on the RTX side. Therefore the control signal is periodically calculated and transferred to the slave node with distributed clock by the EtherCAT network. Also the parallel robot path planning, inverse kinematics and disturbance observer is running on the same side. This guaranteed the synchronization time clock and calculation with same source such as joint space angle and velocity.

In order to investigate the influence and performance of motion control operation in the presence of both kinematical and dynamical inaccuracies, experiments will test on the test setup environment. The experiment setup consists of a parallel robot namely, NT-parallel robot.

First experiments are conducted using measurement taken from the actuator's encoder. Then next experiments are conducted using measurement of the end effector's exact position by vision or laser scanner.

The Ethercat message are transmitted under 10kHz(periodically 100usec) from master pc to all slave devices. The message contain the calculated torque signal w.r.t. joint axis level and individual sensing signals.

Thus these message is important to controlled system for safety and robustness. Our system has safety algorithm for packet drop and message pollution. If the message packet drop occurs, the system safety algorithm detect the emergency and all control signal are stayed on previous value or zero state value. The four parallel robot are able to control by one pc platform.

The proposed fame work is described as below

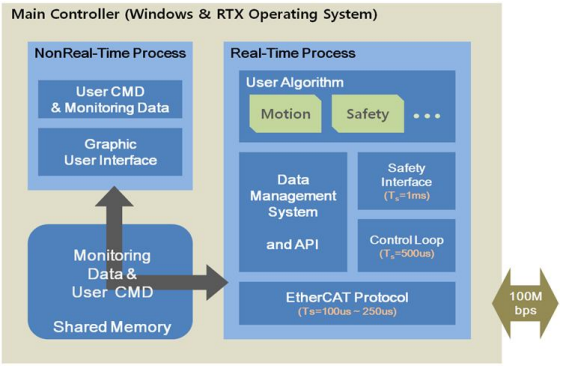


Figure 7. Master control scheme



Figure 8. PC based Master



Figure 9. EtherCAT slave devices and interface board

Overall system are connected by the EtehrCAT and control scheme that ported on the master window pc platform is running on the single-core or multi core. The control performance are guaranteed by the our implementation software platform.

CONCLUSION AND FUTURE WORK

This paper proposed the control scheme of the parallel robot system. For the high speed and precision motion control, we equipped the widow real time operating system, e.x. window RTX, and 100Mbps industrial communication network such as EtherCAT. The comparison between the normal PC-based control system and real time based control system with EtherCAT results will be obtained on the same parallel robot hardware. Also we will prove the feasibility and performance of the proposed control frame work without any change of the parallel robot hardware. This is very important on the parallel robot motion real time tracking, e.g. fast cycle.

ACKNOWLEDGMENT

This work was supported by the Industrial Strategic technology development program, 10041965, Development of a parallel link robot with 6kg payload and 0.43 sec cycling time, funded by the Ministry of Knowledge Economy (MKE), Korea.

REFERENCES

[1] Yangmin Li. “Design and application of a new 3-DOF translational parallel manipulator”, Advanced Robotics and Its Social Impacts, 2007. ARSO 2007. IEEE Workshop on, 9-11 Dec., pp. 1-6(2007)

[2] Khalil, I.S.M.; Golubovic, E.; Sabanovic, A.; ,”High precision motion control of parallel robots with imperfections and manufacturing tolerances,” Emerging Technologies and Factory Automation (ETFA), 2010 IEEE conference on , 13-16 Sept.,vol., no., pp. 1-7(2010)

[3] Il-Seuk Song, “Implementation and analysis of the embedded master for EtherCAT”, Control Automation and Systems (ICCAS), 2010 International Conference on, 27-30 Oct., pp. 2418–2422.(2010)

[4] Rostan, M. “EtherCAT enabled advanced control architecture”, Advanced Semiconductor Manufacturing Conference (ASMC), 2010 IEEE/SEMI, 11-13 July, pp. 39-44(2010)

[5] Prytz, G. “A performance analysis of EtherCAT and PROFINET IRT”, Emerging Technologies and Factory Automation, 2008. ETFA 2008. IEEE International Conference on, 15-18 Sept., pp. 408-415(2008)

[6] E. Tapia, S. Intille and K. Larson, " EUROM ICR0’91 WORKSHOP ON REAL-TIME SYSTEMS," UNIVERSITE PARIS XI PAR I S-0 RSAY, FRANCE, 12-14, June(1991)

[7] C. H. Lee and C. Mavroidis, "PC based control ofand mechatronic systems under MS-windows NT workstation," IEEE/ASME Trans. on Mechatronics,vol.6, no. 3, pp. 311-321,(2001)

[8] W. F. Lages and E. M. Hemerly, "Linux as a software platform for mobile robots," submitted to IEEE Transactions on Software Engineering(200)

★★★