JITTER ANALYSIS OF ETHERCAT MASTER OPEN LIBRARIES IN INDUSTRIAL ROBOT CONTROL SYSTEM

Gruszka L.[*], Kazala R.[**], Michta B.[***], Straczynski P.[****]

Abstract: *This paper presents the comparison of efficiency of control systems based on industrial PC using different open source EtherCAT libraries to communicate with motion control units. The aim of this study is to measure the jitter level in hard realtime application based on Xenomai realtime operating system and application running in user space. Main goal of this research is to find the best solution for positioning six axis manipulator utilized in sorting and packing processes.*

Keywords: EtherCAT, Xenomai, SOEM, IgH Master, Jitter analysis, Industrial robot.

1. Introduction

Rapid development of industrial enterprises is strictly connected with automation and robotization. Replacing human's work by machines is enormously profitable in repetitive processes, serial production and pick and place applications such as packing and the like. Current growth in the field of sensor systems and machine vision provides robots with the ability to detect objects, make decisions and perform much more complicated tasks e.g. sorting or collaboration with environment. Such applications most often urge engineers to utilize six axis robots.

In order to reach peak repeatability of robot's positioning, the communication system between trajectory generator and motion controller has to provide lowest possible delays. Moreover, data exchange should be done with strictly defined sampling time, jitter and delays should be as low as possible or even non-existant. Open source EtherCAT Master libraries e.g. SOEM (Simple Open EtherCAT Master) and IgH Master (EtherLab) can operate in user space as well as on system's kernel level using the real time frameworks such as Xenomai or RTAI (Lee et al., 2011). Designing the most effective system architecture and correct configuration of said libraries require acquaintance with influence of different parameters on time efficiency of motion control system and time deviation associated with it (Cereia et al., 2012). Further in this paper the comparison of jitter in different configurations of communication system for application with high real time requirement is presented.

2. Communication protocols used in motion control

In modern industrial robots' control systems' serial communication busses are the most common form of physical layer. The most widely spread standards in physical layers are: RS-485, CAN, Ethernet (CAN and Ethernet also provide data link layer). Higher layers of communication stack most often utilize dedicated protocols suited for specific application provided by hardware manufacturers, however there is also a wide range of universal protocols e.g. CANopen, EtherCAT, Sercos and the like.

In motion controllers modules designed for robotics produced by companies like Elmo, miControl and so

---

[*]   Lukasz Gruszka, MSc.: Institute of Automatic Control and Robotics , Warsaw University of Technology, ul. Sw. A. Boboli 8, 02-525 Warszaw; PL, lukasz.gruszka.dokt@pw.edu.pl

[**]   Robert Kazala, PhD.: Department of Industrial Electrical Engineering and Automatics, Kielce University of Technology, Aleja Tysiaclecia Panstwa Polskiego 7; 25-314 Kielce; PL, rkazala@tu.kielce.pl

[***]   Bartlomiej Michta, MSc.: Department of Industrial Electrical Engineering and Automatics, Kielce University of Technology, Aleja Tysiaclecia Panstwa Polskiego 7; 25-314 Kielce; PL, bmichta@tu.kielce.pl

[****]   Pawel Straczynski, MSc.: Department of Industrial Electrical Engineering and Automatics, Kielce University of Technology, Aleja Tysiaclecia Panstwa Polskiego 7; 25-314 Kielce; PL, pstraczynski@tu.kielce.pl

on, great share of communication protocols is based on CAN standard, developed by CAN in Automation group (CiA). Said specification was created to meet the requirements set by automation and motion control systems, later on it became a foundation for CANopen. Due to the fact of rapid growth of Ethernet's popularity in the field of automation, process control and data acquisition many protocols designed for Ethernet standard originated from CANopen for example: Ethercat and Powerlink.

Main features of CANopen and it's successors determining their popularity in motion control systems are dedicated specifications e.g. CiA301, CiA402.

In order to provide robot with the ability to move smoothly with lowest possible following error, consecutive trajectory points need to be generated and transmitted to controller in strictly determined time points.

Many different factors may have impact on the sampling time. Therefore, a short time deviation may appear, such occurrence is known as jitter (Delgado et al., 2017).

3. Open-source EtherCAT Master libraries

Communication in EtherCAT is based on master-slave model. In robots' control systems' specialized motion control units work as slave devices. Such controllers perform actions with strict time requirements and provide ability to build real time control systems. In such system architecture role of a master device usually performs an industrial PC with dedicated real time operating system and specialized frameworks and libraries used for communication. In order to properly execute given trajectory master device also has to meet severe real time requirements. Due to the robot's interaction with environment, adopting to new positions, velocities and forces in real time, said factors become essential. These requirements become less severe when robot executes movement along path defined in offline mode.

To choose suitable EtherCAT software from amongst all available, it is necessary to take into consideration it's flexibility and ability to integrate it with many different system e.g. motion controllers, I/O devices. Great part of commercial software implementing EtherCAT functionality is restricted to specified operating system for example Windows (TwinCAT) or to specified device (Hilscher cifX PC card). Moreover, such software is usually closed source what greatly reduces its usefulness for research and academic purposes. Open-source software such as SOEM or EtherLab does not have such flaws, they provide the ability to create control systems based on different devices, operating systems both on user and kernel level.

4. Test system architecture

Experiment was conducted according to the scheme depicted in Fig. 1. Software managing the work of motion controllers and I/O devices was implemented in C++ language as OROCOS module (Orocos Toolchain 2.6).
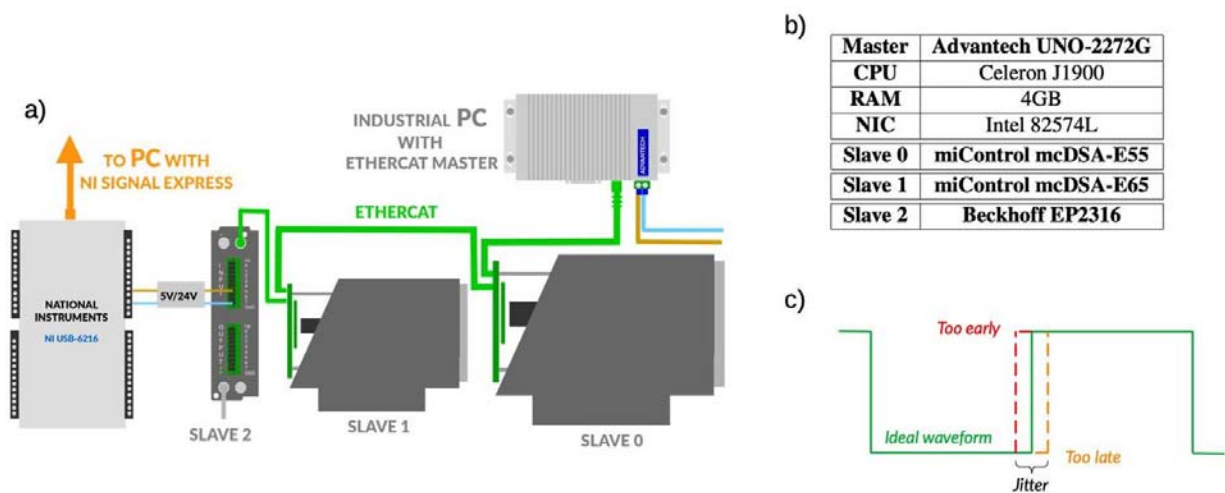

Master	Advantech UNO-2272G
CPU	Celeron J1900
RAM	4GB
NIC	Intel 82574L
Slave 0	miControl mcDSA-E55
Slave 1	miControl mcDSA-E65
Slave 2	Beckhoff EP2316

Fig. 1: Test system a) hardware configuration, b) system components, c) jitter phenomenon.

Orocos Toolchain is a library dedicated for use in real time robotic control systems. It consists of independent modules collaborating with each other. Module responsible for communication with motion controllers gathers data from other modules such as trajectory generators, safety modules and the like. Gathered data is transmitted to slave devices in predefined time intervals.

5. Results

Fig. 2 shows time deviation (jitter) during experiment in which square wave was generated and transmitted to the I/O device with 1ms cycle.
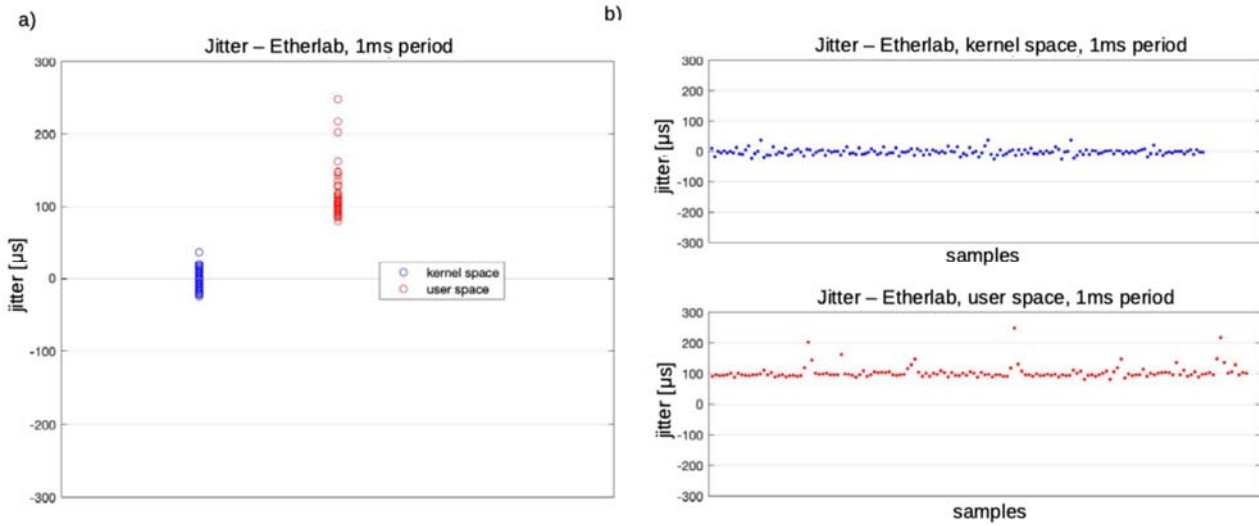
Fig. 2: Result of jitter analysis.

Above charts clearly depict that signal generated by application running on user level has delay somewhere near 100 μs and jitter reaching up to 150 % whereas jitter in application running on kernel level is nearly nonexistent or very low. It means that each cycle of generated signal is delayed by about 10 % at maximum to sampling time.

Tab. 1: Statistic of jitter and latency.

Master	Sampling error [μs]			
	Max	Min	σ	Mean
Etherlab – Xenomai kernel space	37	-25	10.8	-1.6
Etherlab – Linux user space	248	80	22	102.5

Moreover, standard deviation is more than a half lower when running the application on kernel level. Fig. 3 depicts jitter for different configuration of communication system running on Xenomai real time operating system. Figs. 3a and 3b show experiment conducted with EtherLab software, whereas Figs. 3c and 3d present results with SOEM library. Both systems worked on 1 kHz frequency. The aim of this experiment was to study the impact of other real time processes on communication system.

Fig. 3 depicts that both communication systems have similar jitter up to 100 μs when system is heavily loaded with real time processes. Along with decreasing the frequency of additional real time processes system utilizing EtherLab software managed to decrease jitter by nearly a half, whereas system with SOEM software had similar jitter regardless of system's load.
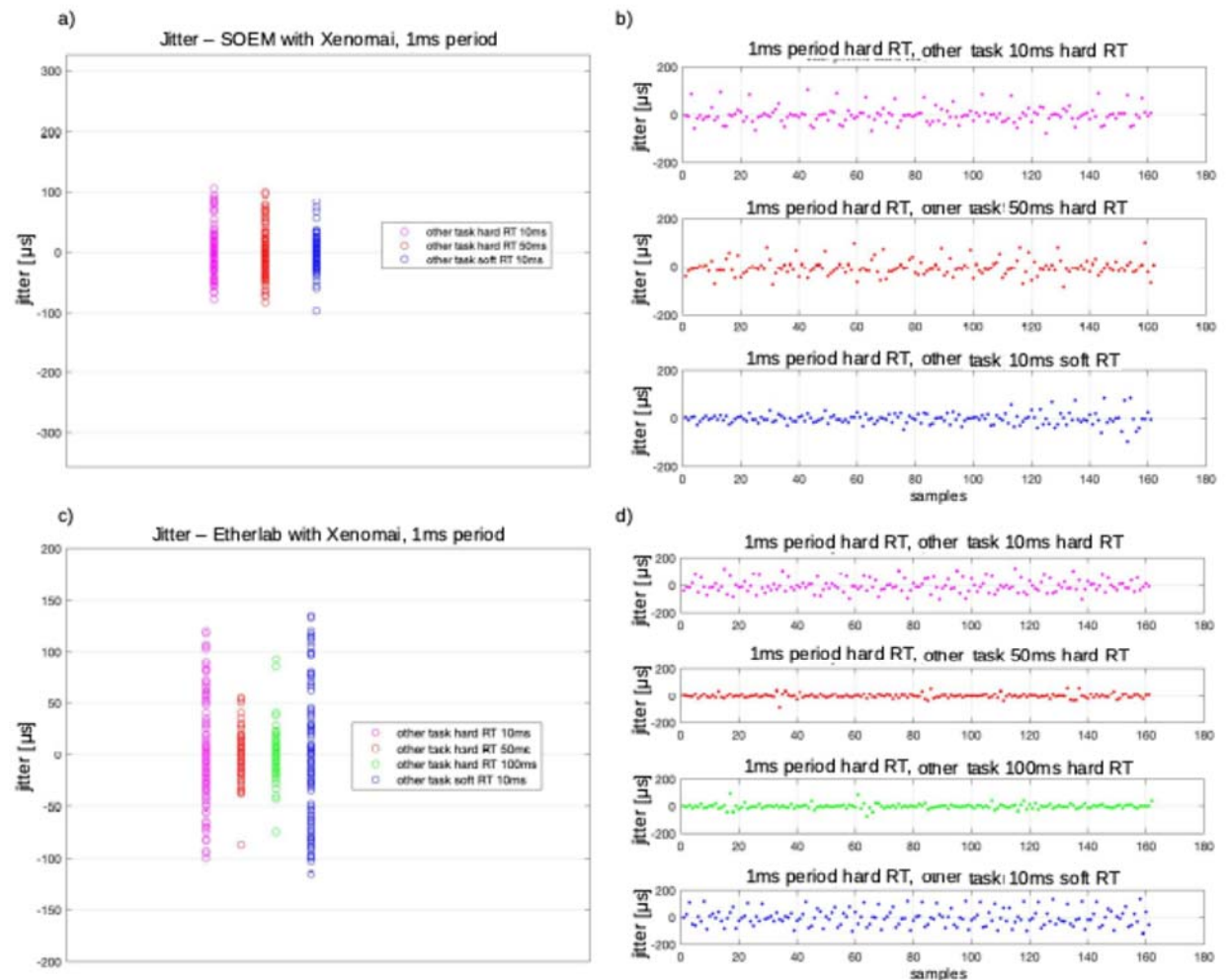
Fig. 3: OROCOS scheduler impact on jitter: a), b) SOME; c), d) EtherLab.

6. Conclusions

The research carried out as part of the project confirmed the possibility of effective use of open implementations of the EtherCAT communication protocol master in the manipulator control system. Based on the experience of attempts to use the SOEM and EtherLab libraries, it has been found that to ensure low time deviations, libraries should be run in the kernel space. Both libraries allow for similar time deviations, however, with longer use, the EtherLab library provided better performance and more configuration options.

References

Cereia M. and Scanzio S. (2012) A user space EtherCAT master architecture for hard real-time control systems, Proceedings of IEEE 17th Int. Conf. on Emerging Technologies & Factory Automation, Krakow, pp. 1-8.

Lee J. K., Song Y. H., Lee S., Lee K. C. and Lee Y. J. (2011) Implementation of multi-axis smart driver system via EtherCAT network based on IEC61800 standard, 11th International Conference on Control, Automation and Systems, Gyeonggi-do, pp. 1871-1874.

Delgado R. and Wook B. (2017) Real-time Servo Control using EtherCAT Master on Real-time Embedded Linux Extensions. International Journal of Applied Engineering Research. 2017:12(21): pp. 11179-11185.