

# Comparative Analysis of EtherCAT Master Open Libraries in Industrial Robot Control System

L. Gruszka<sup>\*</sup>, R. Kazala<sup>\*\*</sup>, B. Michta<sup>\*\*\*</sup>,

## Streszczenie:

Praca przedstawiona analizie porównawcza systemu sterowania wykorzystującego komputer PC oraz protokół EtherCAT z wykorzystaniem otwartych bibliotek. Celem pracy jest zbadanie poziomu jitteru w aplikacji hard realtime wykorzystującej system czasu rzeczywistego Xenomai oraz aplikacji pracującej w przestrzeni użytkownika. Celem badań jest wybranie najlepszego rozwiązania dla zadania pozycjonowania 6-cio osiowego manipulatora wykorzystywanego w automatyzacji procesów sortowania i pakowania.

**Słowa kluczowe:** EtherCAT, Xenomai, SOEM, IgH Master, jitter analysis, industrial robot

## 1. Wprowadzenie

Rosnący rozwój przedsiębiorstw przemysłowych jest związany z ich dynamiczną automatyzacją i robotyzacją. Szczególnie opłacalne dla przedsiębiorców jest zastępowanie człowieka wszędzie tam gdzie wykonywana przez niego praca jest powtarzalna, a więc w szczególności w produkcji wielkoseryjnej czy też w zadaniach typu pick and place takich jak np. pakowanie. Aktualny rozwój w dziedzinie szeroko pojmowanej sensoryki oraz widzenia maszynowego pozwala robotowi na detekcję obiektów, podejmowania decyzji i radzenia sobie z zadaniami mniej standardowymi takimi jak np. sortowanie. Nadzwyczaj zasadne w takich aplikacjach jest zastosowanie manipulatorów 6-cio osiowych których konstrukcja ułatwia tego typu operacje.

Osiągnięcie jak najwyższej powtarzalności pozycjonowania determinuje zastosowanie pomiędzy układem napędowym a jednostką obliczeniową generatora trajektorii systemu komunikacji o możliwie jak najniższych opóźnieniach. Ponadto dane powinny docierać w określonych punktach czasu.

Istniejące otwarte biblioteki Ethercat Master takie jak SOEM (Simple Open EtherCAT Master) oraz IgH Master (Etherlab) pozwalają pracować zarówno w przestrzeni użytkownika jak i na poziomie jądra systemu wspierając frameworki czasu rzeczywistego takie jak Xenomai czy RTAI. Określenie najefektywniejszej architektury systemu i prawidłowa konfiguracja tych bibliotek wymaga znajomości wpływu parametrów na efektywność czasową pracy systemu sterowania ruchem i związane z tym odchylenia czasu. W pracy przedstawiono analizę porównawczą jitteru występującego w systemie komunikacji dla różnych konfiguracji. Porównano wydajności wspomnianych bibliotek w zadaniach o wysokim reżimie czasowym. ?

## 2. Protokoły komunikacyjne użyte w systemie ruchu

W współczesnych systemach sterowania robotów przemysłowych wykorzystywane są szeregowo magistrale komunikacyjne. Wykorzystują one w warstwie fizycznej i łącza danych takie standardy jak RS-485, CAN, Ethernet. W wyższych warstwach stosowane są dedykowane protokoły opracowane przez producentów lub standardowe protokoły takie jak CANopen, Ethercat, Sercos itp.

\* Lukasz Gruszka, MSc.: Institute of Automatic Control and Robotics, Warsaw University of Technology, ul. sw. A. Boboli 8, 02-525 Warszawa; PL, lukasz.gruszka90@gmail.com

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

\*\*\* Bartłomiej Michta, MSc.: Department of Industrial Electrical Engineering and Automatics, Kielce University of Technology, Aleja Tysiąclecia Państwa Polskiego 7; 25-314 Kielce; PL, bmichta@tu.kielce.pl

W modułach napędowych przeznaczonych do wykorzystania w napędach robotów produkowanych przez takie firmy jak Elmo, miControl itp., najczęściej wykorzystywane są protokoły bazujące na specyfikacjach opracowanych przez CAN in Automation group (CiA). Specyfikacje te powstały na potrzeby ułatwienia wykorzystania protokołu CAN w automatyce, a w szczególności w systemach napędowych i były bazą do opracowania protokołu CANopen. W związku z ogromną popularnością standardu Ethernet, w ramach działań pozwalających na ułatwienie zastosowań go w systemach sterowania, opracowano wersje protokołów bazujące na CANopen, takie jak EtherCAT i Powerlink.

Istotną cechą protokołu CANopen i jego pochodnych powodującą częste wykorzystanie w systemach napędowych są dedykowane specyfikacje dla tych systemów, takie jak CiA 301, CiA 402.

Zapewnienie płynnego ruchu ramienia robota i minimalnych błędów trajektorii wymaga generowania i przesyłania informacji o nowych pozycjach w ściśle określonych chwilach czasu.

W systemie sterowania istnieje wiele czynników zakłócających wpływających na dokładność generowania czasu. Powodują one zjawisko zwane jitter, które jest krótkookresowym odchyleniem od oczekiwanych wartości czasu.

### 3. Otwarte biblioteki protokołu Ethercat

Komunikacja w standardzie EtherCAT odbywa się w modelu master-slave. W przypadku systemów sterowania robotów urządzeniami typu slave są specjalizowane kontrolery ruchu. Realizują one swoje funkcje z dość dokładnie określonymi wymaganiami czasowymi i pozwalają na tworzenie systemów sterowania pracujących w czasie rzeczywistym. Funkcje mastera realizowane są najczęściej przez komputery przemysłowe, na których pracuje specjalizowane oprogramowanie i najczęściej system operacyjny czasu rzeczywistego. W celu zapewnienia poprawnej realizacji zadanych trajektorii ruchu robota na mastera nakładane są także dość duże ograniczenia czasowe.

Mają one szczególnie duże znaczenie, kiedy robot wchodzi w interakcje z otoczeniem i wypracowuje nowe pozycje, prędkości lub siły w czasie rzeczywistym.

Dobierając oprogramowanie dla mastera należy wybrać jedną z dostępnych implementacji protokołu EtherCAT. Często wybór konkretnego rozwiązania jest determinowany elastycznością oprogramowania i możliwością jego integracji z różnymi systemami. Część komercyjnych rozwiązań posiada implementacje jedynie na system Windows (TWinCAT) [7531529] albo dedykowane urządzenia (Hilscher cifX PC card), rozwiązania płatne cechują się również kodem zamkniętym uniemożliwiając ingerencje oraz ograniczając możliwości badawcze.

Rozwiązania dystrybuowane jako oprogramowanie open-source, takie jak SOEM czy EtherLab, są pozbawione powyższych wad, dając możliwość pracy oraz badań na różnych platformach sprzętowych, systemach operacyjnych, zarówno w przestrzeni użytkownika jak i na poziomie jądra systemu.

### 4. Architektura systemu testowego

Eksperyment został przeprowadzony zgodnie ze schematem przedstawionym na rysunku 1. Oprogramowanie

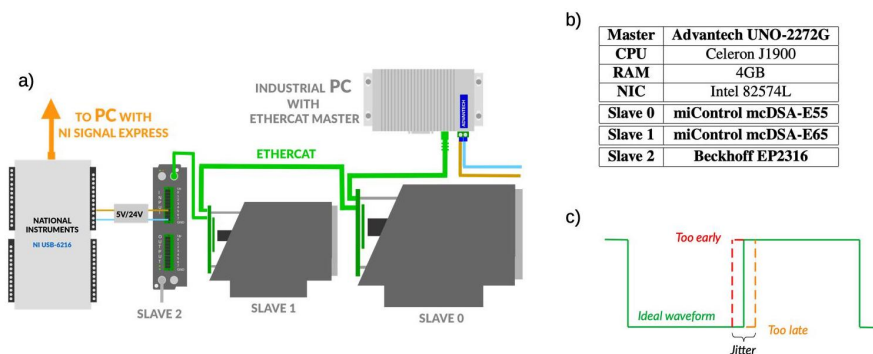


Fig. 1: a) Measuring system, b) Hardware configuration, c) Jitter phenomenon

sterujące prace napędów oraz cyfrowych wejść/ wyjść napisane zostało w języku C++ w postaci modułu OROCOS (Orocos Toolchain 2.6). Orocos Toolchain to biblioteka dedykowana do tworzenia systemów sterujących robotów w czasie rzeczywistym w postaci współpracujących ze sobą komponentów oprogramowania [1]. Moduł sterujący pracą napędów komunikuje się z innymi modułami oprogramowania robota takimi jak generatory trajektorii etc. Oprogramowanie determinuje przesyłanie danych sterujących do urządzeń typu slave w zadanych odstępach czasowych.

## 5. Wyniki

Na rysunku 2 przedstawiono wyniki analizy porównawczej jitteru. Dla zadanego okresu 1ms wygenerowano sygnał prostokątny którego przebieg wpisano na wyjście modułu zdalnych wejść/wyjść.

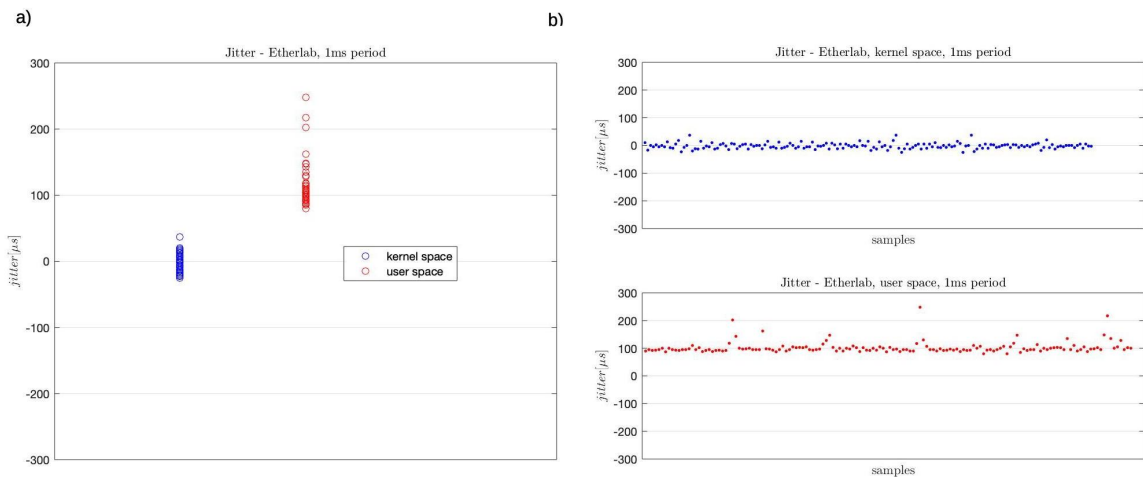


Fig. 2: Result of jitter analysis

Tab. 1: Statistic of the jitter

Master	Jitter [ $\mu s$ ]			
	Max	Min	$\sigma$	Mean
Etherlab in kernel space	37	-25	10,8	-1,6
Etherlab in user space	248	80	22	102,5

Z analizy powyższych wykresów wyraźnie widać, że w przebiegu generowanym w przestrzeni użytkownika, wartość średnia jitteru jest przesunięta w okolice  $100\mu s$  podczas gdy w pracy na poziomie jądra systemu odczytana wartość oscyluje wokół zera. Oznacza to, że każdy cykl generowanego przebiegu jest opóźniony o wartość ok. 10% względem okresu próbkowania.

Ponadto odchylenie standardowe w przypadku zadań realizowanych na poziomie jądra systemu maleje o ponad połowę wartości zmierzonej w przestrzeni użytkownika.

Na rysunku 3 pokazano wyniki zmierzonego jitteru dla różnych konfiguracji z systemem czasu rzeczywistego Xenomai. Rysunek 3 a) i B) przedstawia testy przeprowadzone z oprogramowaniem EtherLab, natomiast rysunek 3 c) i d) z oprogramowaniem SOEM. Oba systemy badano zadając sygnały z częstotliwością 1kHz. Badano wpływ obciążenia systemu innymi zadaniami, które różnicowano ich okresem próbkowania.

Z przedstawionych wykresów można wysnuć wniosek, że przy dużym obciążeniu systemu zadaniami realizowanymi w czasie rzeczywistym, oba systemy zachowują się podobnie a wartość jittera sięga  $100\mu s$ . Natomiast gdy okres realizacji pobocznych zadań zwiększony został do  $50ms$ , odchylenie standardowe jitteru znacząco spada w przypadku z technologią EtherLab. W przypadku SOEM wyraźnej zmiany nie zaobserwowano

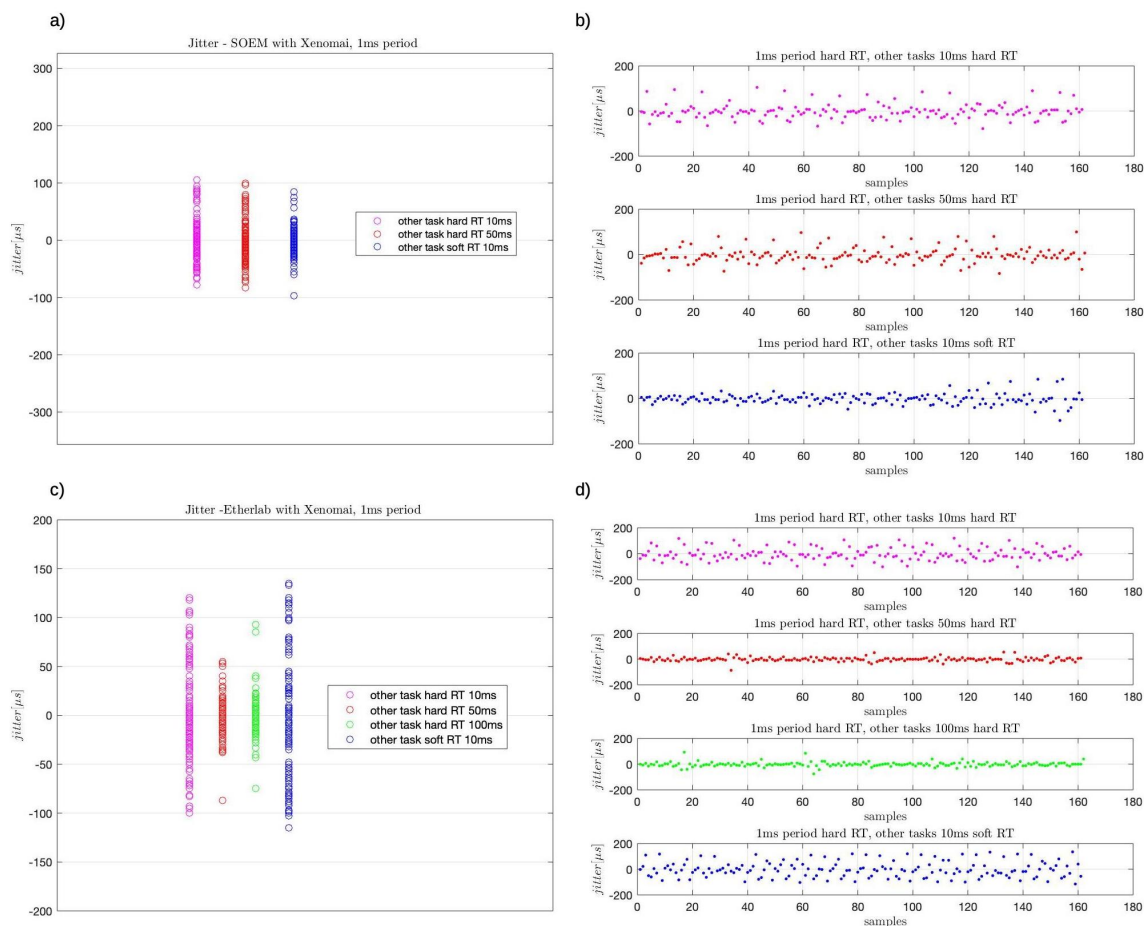


Fig. 3: OROCOS scheduler impact on jitter: a,b) SOEM c,d) EtherLab

## 6. Wnioski

Przeprowadzone w ramach realizacji projektu badania potwierdziły możliwość efektywnego zastosowania otwartych implementacji mastera protokołu komunikacyjnego EtherCAT w systemie sterowania manipulatora. Na bazie doświadczeń związanych z próbami wykorzystania biblioteki SOEM i EtherLab stwierdzono, że w celu zapewnienia niskich odchyleń czasów

Przy czasie komunikacji SOEM

## 7. Oświadczenie

Prace wykonano w ramach projektu: "Opracowanie technologii systemów automatyzacji i robotyzacji procesów technologicznych sortowania i pakowania z wykorzystaniem manipulatorów i chwytaków", realizowanego w ramach Działania 1.1 "Projekty B+R przedsiębiorstw, Poddziałania 1.1.1 Badania przemysłowe i prace rozwojowe realizowane przez przedsiębiorstwa" Programu Operacyjnego Inteligentny Rozwój 2014-2020 na podstawie umowy o dofinansowanie: POIR.01.01.01-00-0566/15 zawartej przez Sorter Michał Ziomek Spółka Jawna z Narodowym Centrum Badań i Rozwoju.