

## SYSTEM STEROWANIA INDUKCYJNYM SILNIKIEM LINIOWYM NA BIEŻNI LINIOWEJ – ANALIZA PROBLEMÓW DZIAŁANIA WYKORZYSTANYCH PROTOKOŁÓW KOMUNIKACYJNYCH

Krzysztof KWASIGROCH<sup>1</sup>, Maciej ŁUKASZEWICZ<sup>2</sup>, Ireneusz MOSOŃ<sup>3</sup>

1. Politechnika Gdańska, Wydział Elektrotechniki i Automatyki  
tel.: 782 017 500 e-mail: krzysztofkwasiagroch@gmail.com
2. Politechnika Gdańska, Wydział Elektrotechniki i Automatyki  
tel.: 791 202 095 e-mail: s160957@student.pg.edu.pl
3. Politechnika Gdańska, Wydział Elektrotechniki i Automatyki  
tel.: 58 347 1485 e-mail: ireneusz.moson@pg.edu.pl

**Streszczenie:** W artykule opisano stanowisko laboratoryjne z silnikiem liniowym na bieżni liniowej znajdujące się w Katedrze Energoelektroniki i Maszyn Elektrycznych na Wydziale Elektrotechniki i Automatyki Politechniki Gdańskiej oraz omówiono, w jaki sposób rozwiązano problemy związane z działaniem wykorzystanych w systemie sterowania protokołów komunikacyjnych. Stanowisko ma charakter dydaktyczny. W skład stanowiska wchodzi indukcyjny silnik liniowy zasilany z przemiennika częstotliwości oraz jego system sterowania, którego głównymi elementami są: sterownik programowalny, dotykowy panel operatorski zintegrowany ze sterownikiem programowalnym oraz moduł rozproszonych wejść/wyjść. Wymiana danych pomiędzy sterownikiem programowalnym a modułem rozproszonych wejść/wyjść jest realizowana z wykorzystaniem protokołu CANopen, pomiędzy sterownikiem programowalnym a panelem operatorskim poprzez sieć Ethernet z protokołem UDP, a pomiędzy panelem operatorskim i modułem komunikacyjnym przemiennika częstotliwości z wykorzystaniem protokołu Modbus RTU. W artykule omówiono problem związany z wymianą danych pomiędzy sterownikiem programowalnym a panelem operatorskim z wykorzystaniem zmiennych sieciowych. Przedstawiono dwa sposoby, dzięki którym możliwe było rozwiązanie problemu błędów w komunikacji z wykorzystaniem zmiennych sieciowych.

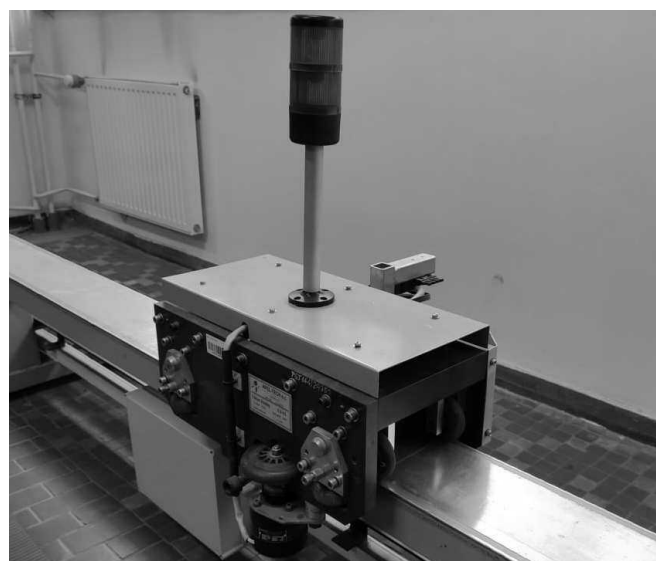
**Słowa kluczowe:** silnik liniowy, sterownik programowalny, panel operatorski, zmienne sieciowe.

### 1. OPIS STANOWISKA LABORATORYJNEGO

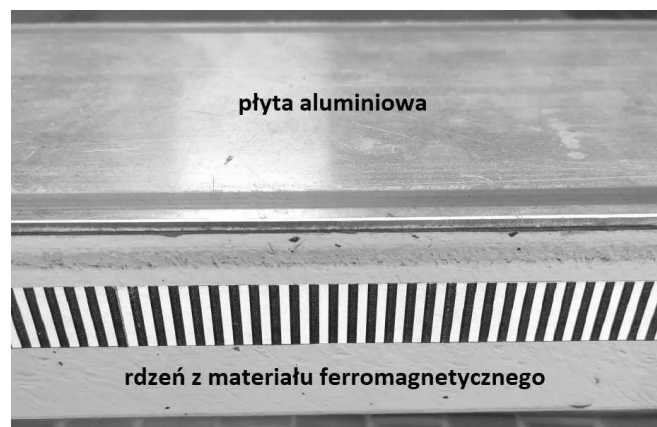
#### 1.1. Silnik liniowy i bieżnia

Urządzeniem wykonawczym na stanowisku laboratoryjnym [1,2,3] jest silnik liniowy SL-5-270 [4,5]. Jest to silnik indukcyjny 3-fazowy jednostronny, zasilany napięciem przemiennym 230/400V o częstotliwości 50Hz. Część pierwotną silnika, nazywaną induktorem, stanowi pakiet blach elektrotechnicznych oraz ułożone w jego żłobkach trójfazowe uzwojenie dwuwarstwowe z cewkami kompensacyjnymi, wykonane z przewodów miedzianych. Silnik zamontowany jest w konstrukcji mechanicznej wózka jezdnych – przedstawionej na rysunku 1 – która umożliwia jego ruch wzdłuż bieżni. Bieżnia wykonana jest z litego rdzenia ferromagnetycznego zapewniającego odpowiedni przepływ strumienia magnetycznego w części wtórnej oraz z płyty z materiału nieferromagnetycznego dobrze przewodzącego prąd elektryczny. Na stanowisku jest to płyta

aluminiowa zamontowana od strony szczeliny powietrznej. Długość bieżni wynosi 18m. Fragment bieżni przedstawiono na rysunku 2. Wzdłuż bieżni rozmieszczonych jest 7 czujników indukcyjnych oraz 2 wyłączniki krańcowe, co pokazano na poglądowym rysunku 3.



Rys. 1. Wózek z silnikiem liniowym na bieżni liniowej



Rys. 2. Fragment bieżni



Rys. 3. Rozmieszczenie czujników indukcyjnych i wyłączników krańcowych wzdłuż bieżni

## 1.2. Zasilanie i system sterowania

Silnik jest zasilany z przemiennika częstotliwości MMX34AA [6]. Sterowanie prędkością silnika jest realizowane w oparciu o metodę U/f. W obwodzie zasilania przemiennika częstotliwości zamontowano filtr sieciowy, który zapewnia ochronę sieci zasilającej przed zakłóceniami w postaci wyższych harmonicznych, których źródłem jest przemiennik częstotliwości.

Głównym elementem sterującym pracą układu jest modułowy sterownik programowalny XC-CPU201-EC512K-8DI-6DO-XV (dalej nazywany XC-CPU201) [7,8]. Posiada on 8 wejść oraz 6 wyjść cyfrowych (binarnych). Jednostka centralna oraz wejścia i wyjścia mają osobne zasilanie. Do zasilania jest wymagany zasilacz 24V DC. Na stanowisku wykorzystano zasilacz EASY500-POW. Do komunikacji sterownika programowalnego z innymi urządzeniami można wykorzystać wbudowane interfejsy: Ethernet/RS232 oraz CANopen.

W celu zapewnienia założonej funkcjonalności stanowiska laboratoryjnego konieczny jest odczyt danych z wielu urządzeń. Sam sterownik programowalny nie posiada wystarczającej liczby wejść i wyjść cyfrowych. Zastosowano więc moduł rozproszonych wejść/wyjść XNE-GWBR-CANOPEN [9], który zamontowano na środku pod bieżnią, aby sumaryczna długość przewodów dochodzących do niego od czujników indukcyjnych i wyłączników krańcowych była jak najmniejsza. Moduł rozproszonych wejść i wyjść składa się z 2 modułów wejść XNE-8DI-24VDC-P (każdy zawiera 8 wejść) oraz jednego modułu wyjściowego XNE-8DO-24VDC-0,5A-P (zawiera 8 wyjść).

Stanowisko laboratoryjne wyposażone jest także w zintegrowany ze sterownikiem programowalnym panel operatorski XV-102-D6-70TWR (dalej nazywany XV-102) [10], który umożliwia obsługę systemu przez operatora. Wszystkie elementy sterowania umieszczone w szafie sterowniczej przedstawionej na rysunku 4. Obwody zasilania i sterowania zostały zabezpieczone wyłącznikami nadprądowymi, zapewniono także ochronę przeciwporażeniową wykorzystując w tym celu wyłącznik różnicowoprądowy. Na elewacji szafy sterowniczej, jak pokazano na rysunku 5, umieszczono panel operatorski, przyciski sterowania oraz lampki sygnalizacyjne.

Zamontowane wzdłuż bieżni czujniki indukcyjne oraz wyłączniki krańcowe umownie dzielą bieżnię na 10 stref. Czujniki indukcyjne są wykorzystywane do określania i kontrolowania pozycji wózka na bieżni, zaś wyłączniki krańcowe mają zabezpieczać przed wyjazdem wózka poza dozwolone strefy ruchu. Na wózku zamontowany jest odbiciowy czujnik optyczny, służący do pomiaru prędkości liniowej wózka. Generacja impulsów następuje w trakcie przemieszczania się wózka, gdy wiązka światła wysyłana przez nadajnik czujnika ulega odbiciu od naklejonej wzdłuż bieżni taśmy pomiarowej z naniesionymi na nią białymi i czarnymi paskami – rysunek 2. Sygnały z czujnika optycznego są podawane na wejście szybkiego licznika sterownika programowalnego XC-CPU201. Do pomiaru prędkości wykorzystano metodę zliczania impulsów w zadanym stałym czasie pomiaru.



Rys. 4. Widok szafy sterowniczej (1 – przemiennik częstotliwości, 2 – sterownik programowalny, 3 – panel operatorski, 4 – zasilacz, 5 – filtr sieciowy)



Rys. 5. Widok elewacji szafy sterowniczej

## 1.3. Struktura komunikacji szeregowej

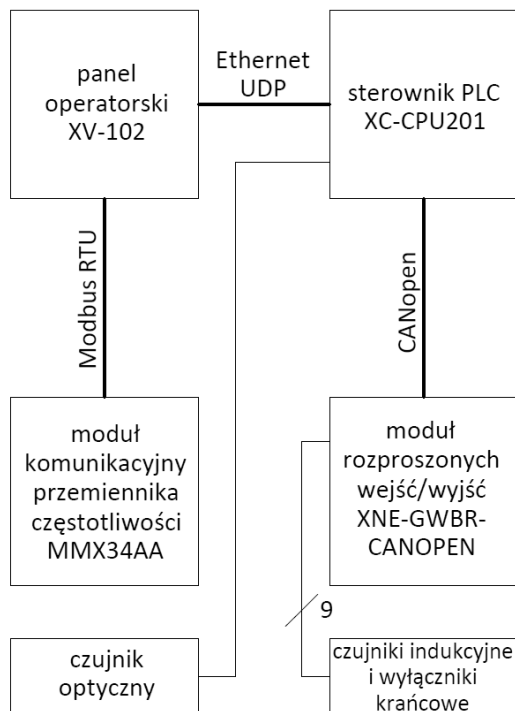
Struktura komunikacji szeregowej w systemie sterowania została przedstawiona na rysunku 6. Wymiana danych pomiędzy urządzeniami zainstalowanymi na stanowisku laboratoryjnym odbywa się przy wykorzystaniu 3 różnych protokołów komunikacyjnych.

Do komunikacji sterownika programowalnego XC-CPU201 z modułem rozproszonych wejść-wyjść XNE-GWBR-CANOPEN wykorzystano sieć CANopen. Sieć ma topologię magistrali, a medium transmisji jest dwużyłowy, ekranowany kabel.

Komunikacja pomiędzy sterownikiem XC-CPU201 a panelem operatorskim XV-102 odbywa się po sieci Ethernet z wykorzystaniem protokołu UDP. Dane wysyłane protokołem UDP są w postaci zmiennych sieciowych, których wartość może być zmieniana w sterowniku programowalnym oraz w panelu operatorskim. Zmienne

sieciowe (z taką samą nazwą i typem zmiennej) muszą być zadeklarowane we wszystkich urządzeniach pomiędzy którymi są one wymieniane.

Komunikaty sterujące pracą przemiennika częstotliwości są wysyłane z panelu operatorskiego XV-102 za pomocą protokołu Modbus RTU. Medium transmisji jest ekranowana skrętka. Urządzenia w tej sieci wykorzystują metodę master/slave dostępu do magistrali. W przypadku stanowiska laboratoryjnego przemiennik częstotliwości to slave, zaś panel operatorski to master. Zaimplementowany w sterowniku programowalnym zintegrowany z panelem operatorskim program wysyła za pomocą protokołu Modbus RTU polecenia i parametry sterowania. Jako interfejs komunikacyjny wykorzystano RS-485.



Rys. 6. Schemat ideowy układu

#### 1.4. Funkcje sterowania

Układ sterowania został tak zaprogramowany, aby wózek z silnikiem liniowym mógł poruszać się po bieźni liniowej w różnych trybach pracy. Zostały przewidziane oraz oprogramowane następujące tryby pracy:

**Sterowanie ręczne** – Do sterowania ręcznego można użyć przyciski na elewacji szafy sterowniczej lub przyciski utworzone na ekranie sterowania ręcznego dotykowego panelu operatorskiego XV-102. Na elewacji przewidziano trzy przyciski do sterowania ręcznego: START LEWO, STOP, START PRAWO. Analogiczne pod względem funkcji przyciski utworzono na ekranie sterowania ręcznego panelu, opisane jako: LEWO, STOP, PRAWO. Operator ma także możliwość zadania częstotliwości napięcia zasilającego silnik.

**Pozycjonowanie** – Tryb ten służy do określenia pozycji wózka na bieźni. Pozycjonowanie powinno być włączane zawsze przy uruchomieniu stanowiska. Jest to konieczne, ponieważ założono, że przy wyłączonym stanowisku możliwe jest ręczne przesunięcie wózka. Operacja pozycjonowania polega na zadaniu ruchu w prawą stronę bieźni; gdy wózek dojedzie do czujnika indukcyjnego lub wyłącznika krańcowego, wówczas system sterowania zna już pozycję wózka i tryb pozycjonowania się kończy.

**Dojazd do czujnika** – Pozwala na dojazd wózka z dowolnej części bieźni do dowolnego czujnika indukcyjnego.

**Jazda cykliczna** – Ten tryb umożliwia jazdę wózka pomiędzy dwoma wybranymi czujnikami indukcyjnymi. Wózek będzie jeździł od jednego wybranego czujnika do drugiego do momentu zatrzymania przez operatora.

**Pokazowy cykl jazdy** – Jest to tryb, w ramach którego wózek porusza się po bieźni w zaprogramowany sposób. Wózek porusza się między czujnikami z różnymi prędkościami. Algorytm pokazowego cyklu jazdy (po wcześniejszym wykonaniu pozycjonowania) jest następujący:

- zadanie częstotliwości 3Hz i dojazd do czujnika 1,
- zadanie częstotliwości 20Hz oraz dojazd do czujnika 6,
- wyłączenie zasilania, przejazd wózka wybiegiem w kierunku czujnika 7 oraz nawrót po dojechaniu do czujnika 7,
- zadanie częstotliwości 14Hz oraz dojazd do czujnika 3,
- nawrót i dojazd do czujnika 4,
- nawrót i dojazd do czujnika 3,
- zadanie częstotliwości 6Hz, nawrót, dojazd do czujnika 5 oraz włączenie migania zielonego sygnalizatora na kolumnie wózka,
- odczekanie 3 sekund, wyłączenie migania zielonego sygnalizatora, dojazd do prawego wyłącznika krańcowego,
- zadanie częstotliwości 3Hz, odczekanie 3 sekund oraz dojazd do czujnika 1.

Sterowanie w ramach każdego z wymienionych trybów pracy realizowane jest z poziomu dotykowego panelu operatorskiego po wywołaniu odpowiedniego ekranu. Wyboru trybu pracy (i odpowiadającego mu ekranu sterowania) dokonuje się na ekranie menu na panelu operatorskim. Większość trybów pracy (za wyjątkiem pozycjonowania) posiada swoją osobną stronę w panelu operatorskim.

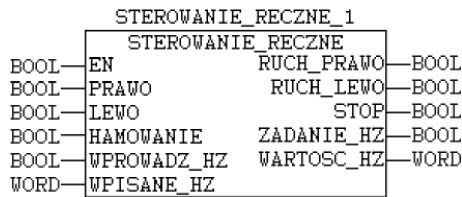
## 2. PROGRAM STEROWANIA I WIZUALIZACJA

### 2.1. Program główny oraz bloki funkcyjne użytkownika

Do napisania programu na sterownik programowalny XC-CPU201 oraz sterownik programowalny zintegrowany z panelem operatorskim XV-102 wykorzystano oprogramowanie easy Soft CoDeSys ver.2.3.9 [11]. Do utworzenia poszczególnych elementów oprogramowania (program główny oraz bloki funkcyjne użytkownika) wykorzystano dwa języki programowania: drabinkowy (LD – ang. Ladder Diagram) oraz tekst strukturalny (ST – ang. Structured Text).

Program główny sterownika został napisany w języku drabinkowym. Czytelną strukturę programu osiągnięto dzięki utworzeniu i wykorzystaniu w programie głównym bloków funkcyjnych użytkownika (BFU). W BFU zostały oprogramowane wszystkie tryby sterowania pracą wózka. Osobne BFU utworzono również do: pomiaru prędkości liniowej, sygnalizacji awarii oraz określania pozycji wózka na bieźni. BFU trybów sterowania wózka posiadają 5 wyjść służących do sterowania pracą przemiennika częstotliwości. Są to 4 wyjścia typu BOOL: zadanie ruchu w prawo, zadanie ruchu w lewo, stop oraz zadanie nowej częstotliwości. Ostatnie wyjście jest typu WORD i służy do zadania nowej wartości częstotliwości. Większość bloków funkcyjnych posiada tylko jedno wejście, którym jest EN (ENABLE) sterujące wykonywaniem BFU. Wyjątkiem jest blok

funkcyjny sterowania ręcznego, pokazany na rysunku 7, ponieważ potrzebuje on informacji z przycisków umieszczonych na elewacji szafy sterowniczej (lub odpowiednich przycisków na ekranie dotykowego panelu operatorskiego).



Rys. 7. Przykładowy blok funkcyjny użytkownika

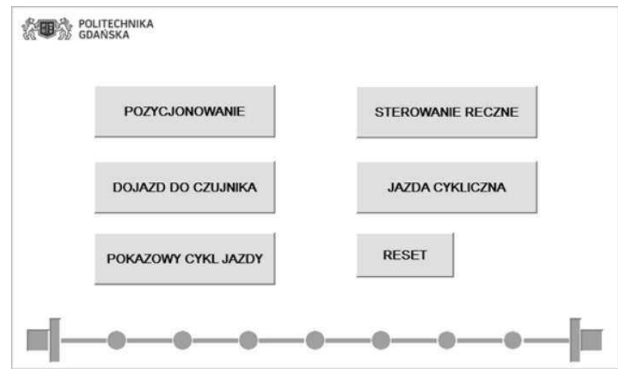
Poprawne działanie programu opiera się na informacji o aktualnej pozycji wózka na bieżni, która określana jest na podstawie sygnałów z czujników indukcyjnych, wyłączników krańcowych oraz kierunku ruchu wózka zadawanego przez przemiennik częstotliwości. Pozycja wózka na bieżni mogłaby być również określana na podstawie liczenia impulsów z czujnika optycznego, przyjmując pozycję wózka przy jednym ze zderzaków za pozycję bazową. W napisanym programie sterowania nie wykorzystano jednak takiej możliwości.

## 2.2. Struktura i ekrany wizualizacji

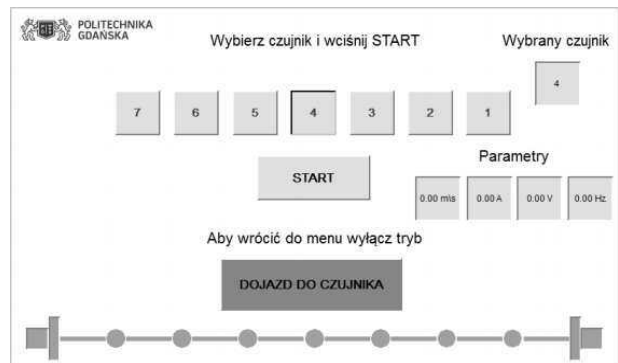
Znajdujący się na stanowisku laboratoryjnym panel operatorski daje możliwość sterowania wózkiem z silnikiem liniowym oraz wizualizacji jego pracy. Do projektowania wizualizacji wykorzystano oprogramowanie Galileo, które służy do programowania paneli operatorskich serii XV firmy Eaton [10]. Opracowana wizualizacja składa się z kilku ekranów. Po włączeniu stanowiska laboratoryjnego na ekranie panelu operatorskiego pojawia się ekran startowy, zaprezentowany na rysunku 8. Po naciśnięciu przycisku START zostaje nawiązana komunikacja sterownika programowalnego panelu operatorskiego z przemiennikiem częstotliwości. Jeśli komunikacja przebiegnie prawidłowo to pojawi się przycisk MENU, którego naciśnięcie spowoduje przejście do menu głównego; jego widok przedstawiono na rysunku 9. W menu głównym istnieje możliwość wyboru kilku trybów pracy wózka. Wybranie większości z nich powoduje przejście do nowego ekranu, za wyjątkiem trybu pozycjonowania. Na każdym ekranie widoczna jest również grafika przedstawiająca pozycję wózka na bieżni. Podczas pracy silnika wyświetlane są parametry takie jak: prędkość liniowa, prąd, napięcie oraz częstotliwość. Na rysunku 10 przedstawiono ekran na panelu operatorskim, który dotyczy trybu dojazdu do czujnika.



Rys. 8. Widok ekranu startowego



Rys. 9. Widok ekranu menu



Rys. 10. Widok ekranu trybu dojazdu do czujnika

## 3. URUCHOMIENIE STANOWISKA I TESTY

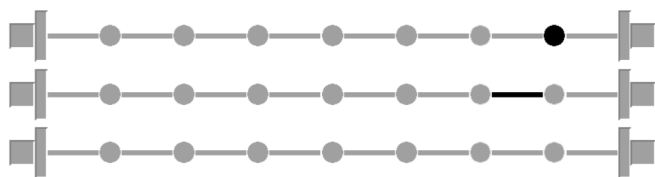
### 3.1. Testy poprawności działania oprogramowania

Finalnie uruchomienie stanowiska i przeprowadzone testy poprawności działania opracowanego oprogramowania i wizualizacji zakończyły się powodzeniem. Z wynikiem pozytywnym przeprowadzono testy poprawności realizacji algorytmów sterowania we wszystkich przewidzianych trybach pracy. Jednakże uruchomienie i testy nie przebiegały bezproblemowo. W czasie ich przeprowadzania zauważono błędy, związane z aktualizacją zmiennych sieciowych, a sposób ich wyeliminowania nie był od początku oczywisty.

Błędy występowały przy domyślnych ustawieniach parametrów pracy sterownika oraz parametrów aktualizacji zmiennych sieciowych. Objawiały się tym, że nie zawsze zmienne sieciowe były aktualizowane. Przykładowo, system sterowania odczytuje sygnały z czujników indukcyjnych, wyłączników krańcowych oraz przycisków znajdujących się na elewacji szafy sterowniczej i na ekranie panelu operatorskiego. Niestety, nie zawsze te informacje były poprawnie odbierane.

Nieprawidłowości związane z komunikacją zauważono również w trakcie wyświetlania na panelu operatorskim aktualnej pozycji wózka na bieżni. Algorytm określania pozycji wózka bazuje na informacji z czujników indukcyjnych i wyłączników krańcowych oraz na ostatnim zadanym kierunku jazdy wózka. Na rysunku 11 górna grafika przedstawia położenie wózka nad pierwszym czujnikiem indukcyjnym, a grafika środkowa przedstawia położenie wózka w strefie pomiędzy pierwszym i drugim czujnikiem indukcyjnym. Problemy z wizualizacją pozycji występowały, gdy wózek przejeżdżał ze strefy do strefy. W programie sterowania znana była prawidłowa pozycja wózka, natomiast w wizualizacji (zmienna sieciowa)

czasami nie była wyświetlana żadna pozycja wózka. Przypadek taki pokazano na dolnej grafice na rysunku 11.



Rys. 11. Grafika przedstawiająca pozycje wózka na bieźni

### 3.2. Testy komunikacji szeregowej w sieci Ethernet z protokołem UDP oraz sposoby eliminacji problemów komunikacyjnych

W celu ustalenia źródła opisanych powyżej problemów przeanalizowano działanie sieci komunikacyjnych. Wartości wysyłane siecią CANopen przesyłane były bezbłędnie oraz bez widocznego opóźnienia. W przypadku komunikacji panelu operatorskiego z przemiennikiem częstotliwości początkowo zdarzały się sytuacje, że dane były wysyłane ze zwłoką lub nie były wysyłane wcale. Spowodowane to było błędami w ustawieniach wartości parametrów bloku funkcyjnego odpowiadającego za komunikację w sieci Modbus RTU. Po dokonaniu korekt wartości parametrów komunikacja w sieci Modbus RTU przebiegała bez problemów.

Podgląd programów w poszczególnych trybach pracy pozwolił zauważyć występujące błędy w komunikacji sterownika programowalnego z panelem operatorskim. W celu sprawdzenia poprawności transmisji danych poprzez sieć Ethernet z wykorzystaniem protokołu UDP napisano program testowy realizujący przesłanie tablicy 300 zmiennych sieciowych typu INT (liczby całkowite) ze sterownika programowalnego do panelu operatorskiego. Przesył danych podzielono na 5 etapów inicjowanych sygnałem zewnętrznym (jeden z przycisków na elewacji szafy sterowniczej). Pierwsze cztery naciśnięcia przycisku inicjowały przesłanie kolejno 50 elementów tablicy, a piąte naciśnięcie przycisku inicjowało przesłanie 100 ostatnich elementów tablicy zmiennych sieciowych. Test powtórzono wielokrotnie. Za każdym razem ok. 10% przesłanych danych było błędnych.

Założono, jak się później okazało, słusznie, że problemy wynikają z domyślnych wartości parametrów pracy sterownika oraz parametrów aktualizacji zmiennych sieciowych.

W odróżnieniu od systemów operacyjnych wielu sterowników programowalnych system operacyjny sterownika XC-CPU201 jest systemem wielozadaniowym. Oznacza to, że wiele zadań sterowania może być realizowanych praktycznie w tym samym czasie. W rzeczywistości czas jest dzielony na wykonywanie poszczególnych zadań. Podział czasu na wykonywanie poszczególnych zadań zależy od przyporządkowanego priorytetu zadania oraz czasu cyklu ustawionego dla tego zadania. W tym przypadku określenie „czas cyklu” [8] należy rozumieć jako przedział czasu lub okres, analogicznie jak dla okresowego wykonywania programu sterowania, czyli czas po którym rozpoczyna się kolejne wykonywanie kodu programu realizującego dane zadanie sterowania.

W systemie sterowania silnikiem liniowym, sterownik programowalny XC-CPU201 realizuje jedno zadanie sterowania, a domyślna wartość jego czasu cyklu wynosi 10ms. Zmienne sieciowe są przesyłane w pakietach

wówczas, gdy następują zmiany ich wartości, ale z ustawionym minimalnym przedziałem czasu pomiędzy przesłaniami wynoszącym 20ms. Tak domyślnie ustawione parametry sterownika oraz przesyłania zmiennych sieciowych powodują, że wartość zmiennej utrzymująca się tylko przez jeden cykl nie zawsze przypisze się do zmiennej sieciowej. Aby zmienne sieciowe poprawnie się przypisały, to ich aktualizacja musi następować w cyklu, w którym następuje przypisanie wartości do tych zmiennych.

W związku z powyższym zaproponowano, zastosowano i przetestowano dwa sposoby rozwiązania problemu pojawiania się błędnych wartości przesyłanych zmiennych sieciowych.

Pierwszy ze sposobów polega na zastosowaniu timerów ustawionych na czas umożliwiający przypisanie odpowiedniej wartości do zmiennej sieciowej przez co najmniej kilka cykli programu.

Drugi ze sposobów polega na zadaniu takiego czasu cyklu, aby był on dłuższy od czasu aktualizacji zmiennych sieciowych. Wykonano opisane wcześniej testy polegające na przesłaniu 300 zmiennych sieciowych, ale dla zadanych dłuższych czasów cyklu wynoszących 20ms i 30ms. Dla 20ms występowało ok. 1% błędnych wartości przesyłanych zmiennych sieciowych. Dla czasu cyklu równego 30ms nie było już żadnych błędów w wartościach przesyłanych zmiennych sieciowych.

## 5. PODSUMOWANIE

Modernizacja stanowiska laboratoryjnego z silnikiem liniowym na bieźni liniowej w zakresie sprzętowym i programowym zakończyła się powodzeniem. Stanowisko działa poprawnie we wszystkich przewidzianych i zrealizowanych trybach pracy [1]. Stanowisko ma charakter dydaktyczny. Jest wykorzystywane jako dodatkowe ćwiczenie laboratoryjne w ramach przedmiotu „Sterowniki programowalne” na studiach I stopnia na kierunku Elektrotechnika. Może być również wykorzystane przy realizacji tematów kolejnych prac dyplomowych, a także w ramach przedmiotu „Przemysłowe sieci informatyczne”.

Problemy w komunikacji szeregowej pomiędzy sterownikiem programowalnym i panelem operatorskim zostały rozpoznane i rozwiązane. Z dwóch zaproponowanych sposobów rozwiązania tych problemów oba okazały się skuteczne. W przypadku pierwszego sposobu zmiany dotyczą oprogramowania, natomiast w przypadku drugiego sposobu – odpowiedniego zadania parametrów i są szybsze do wykonania. Należy jednak zauważyć, że sposób wydłużenia czasu cyklu może być zastosowany tylko wówczas, gdy nie spowoduje pogorszenia funkcjonowania sterowanego układu. Gdyby zwiększenie czasu cyklu, skutkujące wydłużeniem czasu reakcji systemu sterowania, nie było możliwe, należałoby, przy nie zmienionym czasie cyklu, zmniejszyć wartość parametru odpowiadającego czasowi aktualizacji zmiennych sieciowych.

Możliwa jest dalsza rozbudowa sprzętowa i programowa stanowiska z silnikiem liniowym na bieźni liniowej. Szczególnie przydatna wydaje się możliwość wykorzystania impulsów z taśmy pomiarowej nie tylko do obliczania prędkości liniowej silnika, ale także do określania pozycji wózka silnika na bieźni (niezależnie od informacji z czujników indukcyjnych i wyłączników krańcowych).

Porównanie tak uzyskanych informacji o położeniu wózka mogłoby stanowić dodatkową informację diagnostyczną.

## 6. BIBLIOGRAFIA

1. Kwasigroch K., Łukaszewicz M.: Modernizacja stanowiska laboratoryjnego z silnikiem liniowym indukcyjnym SL-5-270. Praca dyplomowa inżynierska, Politechnika Gdańska, Gdańsk 2018.
2. Ellwart M., Szczepański M.: Modernizacja stanowiska oraz sterowanie z pomiarem prędkości cyklami jazdy wózka silnika liniowego na bieżni. Praca dyplomowa inżynierska, Politechnika Gdańska, Gdańsk 2012.
3. Miszkiewicz P., Sudoł M., Zieliński P.: Modernizacja stanowiska laboratoryjnego oraz opracowanie programu sterowania z wizualizacją silnikiem liniowym na bieżni liniowej. Praca dyplomowa inżynierska, Politechnika Gdańska, Gdańsk 2011.
4. Silniki liniowe. Ośrodek Badawczo Rozwojowy Maszyn Elektrycznych EMA-KOMEL, Lz. 48/74.
5. Mosoń I.: Sterowniki programowalne – zagadnienia wybrane. Politechnika Gdańska, Wydział Elektrotechniki i Automatyki, Gdańsk 2005.
6. Randermann J.: Frequency converter M-MAX. Quick Start Guide. AWB8230-1604en. Bonn 2009.
7. Wiegand R.: Modular PLC XC-CPU201-...(-XV), XC-CPU202-...-XV. User Manual. MN05003001Z-EN. Bonn 2000.
8. Roersch P.: Modular PLC XC-CPU201-...(-XV). Hardware, Engineering and Functional Description. AWB2724-1491GB. Bonn 2005.
9. Jahn M.: User Manual XI/ON. MN05002005Z-EN, St. Gallen 2011.
10. Pierwsze kroki z panelami Operatorskimi serii XV. NA131PL 01/2011.
11. Albrecht W.: PLC Programming. XSoft version 2.3. AWB2700-1437GB, Bonn 2003.

## CONTROL SYSTEM OF THE LINEAR INDUCTION MOTOR ON THE LINEAR TRUCK – ANALYSIS OF PROBLEMS OF IMPLEMENTED COMMUNICATION PROTOCOLS PERFORMANCE

In the paper a laboratory stand with the linear motor on the linear truck operating in the Department of Power Electronics and Electrical Machines on the Faculty of Electrical and Control Engineering in Gdansk University of Technology as well as the method how the problems related to operation of communication protocols implemented in the laboratory stand control system have been solved are described. The laboratory stand is of a didactic character. The stand consists of an induction linear motor supplied from a frequency converter and the control system, main components of which are: a programmable controller, an operator touch panel integrated with a programmable controller and a module of remote inputs/outputs. Data exchange between the programmable controller and the module of remote inputs/outputs is realized with the use of CANopen protocol, between the programmable controller and the operator panel through the Ethernet network with the use of UDP protocol, and between the operator panel and the communication module of the frequency converter with the use of Modbus RTU protocol. In the paper a problem related to data exchange between the programmable controller and the operator panel with the use of network variables has been described. Two methods, thanks to which it was possible to solve the problem of errors in communication with the use of network variables, have been presented.

**Keywords:** linear motor, programmable controller, operator panel, network variables.