

WYDZIAŁ INŻYNIERII BIOMEDYCZNEJ
POLITECHNIKA ŚLĄSKA

ROZPRAWA DOKTORSKA

TEMAT:

MODULARNY SYSTEM STEROWANIA ROBOTA REHABILITACYJNEGO
NARZĄDÓW RUCHU ZE SPRZĘŻENIEM SIŁOWYM

AUTOR:

mgr inż. Andrzej Michnik

PROMOTOR: prof. dr hab. inż. Robert Michnik

OPIEKUN POMOCNICZY: dr hab. inż. Janusz Wróbel

Podziękowania

Na wstępie, chciałbym serdecznie podziękować mojemu promotorowi Profesorowi Robertowi Michnikowi za nieocenione wsparcie merytoryczne w przygotowaniu niniejszej pracy doktorskiej, a także chciałbym podziękować mojemu opiekunowi Doktorowi Habilitowanemu Januszowi Wróblowi, koleżankom i kolegom z mojego Instytutu, którzy mnie wspierali podczas prac prowadzonych w ramach tego projektu doktorskiego. Również chciałbym podziękować mojej żonie Beacie i synom za wyrozumiałość i wsparcie w tym absorbującym procesie, jakim było pisanie pracy doktorskiej.

SPIS TREŚCI

Wstęp.....	7
1. Przegląd robotów stosowanych w rehabilitacji kończyn górnych	9
2. Systemy sterowania spotykane w robotyce	23
3. Cel i zakres pracy	25
4. Opis systemu sterowania robota rehabilitacyjnego	27
4.1. Prototyp robota rehabilitacyjnego ARM-200	30
4.2. Moduł Master	32
4.3. Moduł Przekazników Bezpieczeństwa	33
4.4. Moduł Slave.....	34
4.5. Moduł Ekspandera.....	36
4.6. Moduł Hamulca Dynamicznego	37
4.7. Moduł sEMG	38
4.8. Czujnik siły.....	39
4.9. Czujnik pozycji bezwzględnej.....	40
4.10. Magistrala danych.....	42
4.11. Automatyczne adresowanie modułów Slave	45
4.12. Aplikacja sterująca robotem	47
4.13. Opis działania oprogramowania modułu Master	52
4.14. Opis działania oprogramowania modułu Slave	55
4.15. Protokół komunikacji	74
4.16. Rejestracja danych przez system sterowania robota.....	97
5. Walidacja działania systemu sterowania.....	99
5.1. Walidacja układów pomiarowych systemu sterowania.....	99
5.1.1. Walidacja układu pomiarowego siły	99
5.1.2. Walidacja układu pomiarowego prędkości kątowej.....	101
5.1.3. Walidacja układu pomiarowego kąta bezwzględnego	103
5.1.4. Badanie działania układu pomiaru prądu zasilania napędu	104
5.1.5. Badanie działania układu pomiaru napięcia zasilania napędu	106

5.2. Analiza wpływu zastosowania systemu operacyjnego nie czasu rzeczywistego (ang.: non-real-time operating system).....	108
5.3. Badanie szybkości komunikacji aplikacji sterującej z robotem.....	113
5.4. Badanie zachowania systemu sterowania w warunkach braku komunikacji z pojedynczym modułem Slave	114
5.5. Testowanie działania robota w pełnej konfiguracji.....	115
5.6. Badanie węzła robota ARM-200 w trybie śledzenia ruchu.....	119
5.7. Badanie ilościowe węzła robota ARM-200 w trybie śledzenia ruchu	121
5.8. Wyznaczenie progów nieczułości sił dla robota ARM-200.....	123
5.9. Walidacja trybu śledzenia ruchu z progami nieczułości	125
6. Podsumowanie i wnioski.....	127
7. Kierunki dalszych badań	129
Streszczenie.....	131
Summary	133
Literatura	135
Spis rysunków	139
Spis tabel	145

WSTĘP

Tematem niniejszej pracy doktorskiej jest przedstawienie opracowanego systemu sterowania robota rehabilitacyjnego, a także przedstawienie metodologii projektowania i walidacji opracowanych rozwiązań.

Żyjemy w starzejącym się społeczeństwie. Według danych Narodowego Spisu Powszechnego przeprowadzonego w roku 2021, w ciągu poprzednich 10 lat polskie społeczeństwo postarzało się średnio o 4 lata. Wyniki spisu ujawniły również wzrost liczby osób deklarujących niepełnosprawność biologiczną, o wartość 26% w porównaniu do poprzedniego spisu w roku 2011 [1]. Z danych systemu EKSMOoN (Elektroniczny Krajowy System Monitorowania i Orzekania o Niepełnosprawności), które zostały udostępnione w raporcie Rządu RP na temat podejmowanych działań na rzecz realizacji postanowień „Karty Praw Niepełnosprawnych” [2, s. 10] wynika, że upośledzenie narządu ruchu jest w Polsce najliczniejszą przyczyną niepełnosprawności i stanowi prawie 31% przypadków. Taki stan rzeczy wynika z wydłużenia czasu życia i niskiej diety a także z chorób cywilizacyjnych, będących wynikiem ograniczania aktywności ruchowej, stresu oraz urazów powstałych w wyniku wypadków komunikacyjnych. Wymienione czynniki wraz z ograniczoną liczbą rehabilitantów, prowadzą do stanu, w którym wielu pacjentów staje się osobami niesamodzielnymi. Jedną z dróg poprawy tej sytuacji może być wykorzystanie robotów wyręczających rehabilitanta od monotonnej i wyczerpującej pracy fizycznej, jednocześnie pozwalającej na realizację treningów prowadzonych z wieloma pacjentami w tym samym czasie [3]. Dodatkowo, rehabilitant korzystający z robota uzyskuje narzędzie diagnostyczne, ponieważ czujniki robota mogą np. zmierzyć zakresy ruchomości w stawach, pobudzenie elektryczne mięśni (sEMG) czy też siłę wybranych mięśni. Patrząc na rozwój rozwiązań robotycznych [3], można założyć, że w niedalekiej przyszłości spopularyzują się również rozwiązania tele-rehabilitacyjne wykorzystujące roboty. Niedawny czas pandemii COVID-19, udowodnił dużą atrakcyjność tego typu rozwiązań.

W pracy przedstawiono założenia i opis realizacji systemu sterowania robota umożliwiającego wykonanie złożonych ruchów wielopłaszczyznowych biernych, czynnych i czynnych z oporem o zakresie oraz dynamice zbliżonej do aktywności naturalnej. Ważnym elementem pracy jest analiza wymagań funkcjonalnych i wymagań bezpieczeństwa, które powinny być spełnione w przypadku urządzenia medycznego.

1. PRZEGLĄD ROBOTÓW STOSOWANYCH W REHABILITACJI KOŃCZYN GÓRNYCH

Rozwój metod diagnostycznych i leczniczych umożliwił ratowania życia i zdrowia w przypadkach bardzo trudnych, które jeszcze niedawno kończyły się zgonem. Jednocześnie w wielu przypadkach, podczas doraźnej interwencji medycznej pacjentowi nie zawsze udaje się przywrócić utracone funkcje. Dopiero w procesie rehabilitacji pacjent uzyskuje możliwość odzyskania utraconych funkcji. Rehabilitacja w obecnych czasach jest elementem składowym skutecznego leczenia, realizowanym interdyscyplinarnie, często obejmując współpracę wielu specjalistów, takich jak fizjoterapeuci, lekarze, psychologowie, dietetycy i terapeuci zajęciowi. Celem jest kompleksowe podejście do pacjenta, uwzględniające zarówno aspekty fizyczne, jak i psychiczne jego zdrowia.

Kończyny górne odgrywają kluczową rolę w funkcjonowaniu człowieka, stanowiąc złożony narząd ruchu o wielu stopniach swobody. Strukturalnie można je podzielić na obręcz kończyny górnej (zwaną również obręczą piersiową) oraz część dolną kończyny górnej. Obręcz kończyny górnej obejmuje obojczyk oraz łopatkę, natomiast część dolna składa się z ramienia, które poprzez staw łokciowy łączy się z przedramieniem. Przedramię, z kolei, łączy się z ręką za pośrednictwem stawu nadgarstkowego. [4]. Dysfunkcje kończyny górnej, wynikające z przebytych urazów lub chorób, mogą istotnie ograniczyć lub całkowicie uniemożliwić realizację wielu podstawowych funkcji, z których zdrowy człowiek zazwyczaj nie zdaje sobie sprawy, takich jak np.:

- chwytanie, trzymanie i precyzyjne manipulowanie przedmiotami, umożliwiające np. pisanie, rysowanie lub inne prace manualne;
- komunikacja niewerbalna, gestykulacja, wyrażanie emocji;
- wykonywanie zadań zawodowych, obsługa maszyn, prowadzenie pojazdów, przygotowywanie posiłków, sprzątanie;
- utrzymanie higieny osobistej, w tym: mycie, ubieranie, czesanie;
- balansowanie kończynami podczas chodzenia lub biegania, podnosi efektywność energetyczną a także zmniejsza ryzyko upadku w wyniku utraty równowagi [5], szczególnie na nierównym terenie;

- kreatywne wykonywanie działań twórczych takich jak: gra na instrumentach, malowanie, rzeźbienie;
- a także udział w aktywnościach sportowych takich jak np.: pływanie, badminton, koszykówka, siatkówka.

Jak wynika z wyżej przedstawionej listy, dysfunkcja kończyny lub obu kończyn górnych, może bardzo negatywnie wpłynąć na wiele aspektów życia codziennego. Z tego względu, kluczowe znaczenie mają techniki rehabilitacyjne, które umożliwiają częściowo lub całkowicie przywrócić utracone funkcje. Przywrócenie sprawności kończyny górnej może wymagać rehabilitacji ortopedycznej lub neurologicznej. Rehabilitacja ortopedyczna jest wymagana po urazach lub stanach chorobowych układu kostnego, stawowego lub mięśniowego. Natomiast rehabilitacja neurologiczna ma na celu poprawę sprawności pacjentów z deficytami ruchowymi i chorobami neurologicznymi. Rehabilitacja neurologiczna wymaga często udziału fizjoterapeuty, psychologa, logopedy i innych specjalistów. Główną przyczyną niepełnosprawności osób dorosłych w Polsce są udary mózgu [6], po przebyciu których wymagane jest szybkie wdrożenie procedur rehabilitacyjnych, a czas jest kluczowym parametrem wpływającym na skuteczność rehabilitacji. Zaleca się wdrożenie procesu wczesnej rehabilitacji zaraz po ustabilizowaniu klinicznego pacjenta z uwzględnieniem deficytów neurologicznych (ruchowych i czuciowych), zaburzeń funkcji poznawczych, zaburzeń behawioralnych i emocjonalnych. Zaleca się wdrożenie ćwiczeń czynnych, jeżeli stan pacjenta na to pozwala lub ćwiczeń biernych, które nie przynoszą tak wyraźnych efektów jak ćwiczenia czynne, ale również w istotny sposób wpływają na układ mięśniowo-szkieletowy, zmniejszając ryzyko zapalenia płuc i zakrzepicy żył głębokich. Następnie, po zakończeniu hospitalizacji, należy wdrożyć dalszy proces rehabilitacji poudarowej. Najlepsze wyniki rehabilitacji poudarowej obserwuje się w pierwszych 12 tygodniach po wystąpieniu udaru [7]. Celem kompleksowej rehabilitacji, obejmującej rehabilitację początkową (na oddziale neurologicznym) i późniejszą w specjalizowanym ośrodku rehabilitacyjnym, jest umożliwienie pacjentowi powrotu do codziennego życia oraz uzyskania przez niego niezależności i samodzielności w wykonywaniu podstawowych czynności życiowych.

Kluczową cechą układu nerwowego, dla skutecznej rehabilitacji neurologicznej, jest tzw. plastyczność mózgu, nazywana również neuroplastycznością. Definiuje się ją jako zdolność do przebudowy połączeń układu nerwowego człowieka na skutek oddziaływania

bodźców ze środowiska zewnętrznego. W przypadku rehabilitacji, mechanizm neuroplastyczności umożliwia rozszerzenie sprawnie funkcjonujących obszarów mózgu o nowe funkcje oraz na tworzeniu nowych połączeń neuronalnych odpowiedzialnych za kontrolę ruchu w porażonych częściach ciała [8, s. 14–15]. Podstawą nowoczesnej rehabilitacji jest kinezyterapia, natomiast u chorych z niedowładem połowicznym po udarze mózgu stosuje się metody, których celem jest uzyskanie jak największej samodzielności i niezależności w wykonywaniu czynności życia codziennego [9]. Do najpopularniejszych i jednocześnie najskuteczniejszych metod rehabilitacyjnych, wykorzystujących neuropatyczność mózgu można zaliczyć:

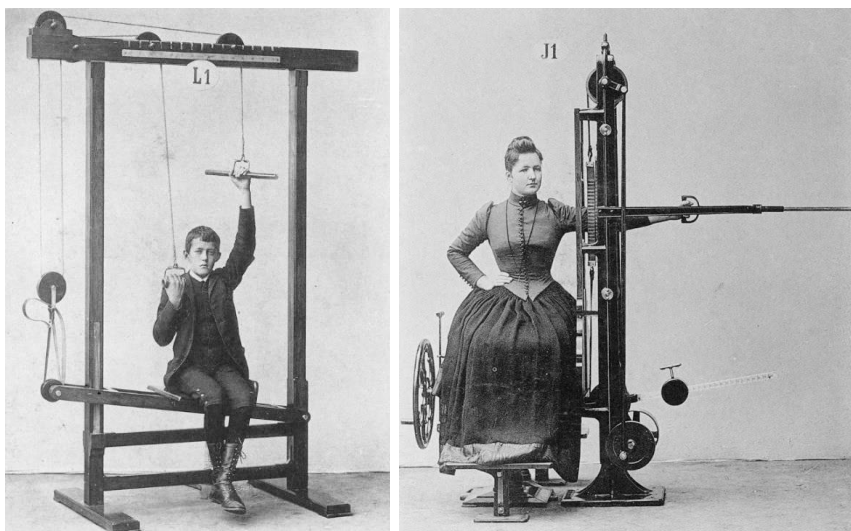
- metodę proprioceptywnego torowania
(PNF – proprioceptive neuromuscular facilitation),
- metodę Bobathów
(NDT – neurodevelopmental treatment),
- metodę wymuszania ruchu
(CIT – constraint-induced movement therapy).
- Stosowane są również metody: Brunnstorm, Rood i Jacobsona [8, s. 15].

Metoda PNF polega na odtwarzaniu z pacjentem złożonych wielopłaszczyznowych sekwencji ruchowych kończyny, tak zwanych wzorców ruchowych. Wykonywanie ćwiczeń w tej metodzie wymaga ogromnego zaangażowania fizjoterapeuty powtarzającego wielokrotnie z pacjentem zadane wzorce ruchowe co stanowi ograniczenie dostępności do odpowiednio częstych sesji rehabilitacyjnych dla pacjentów [10].

Rozwiązaniem pozwalającym na poszerzenie dostępu do rehabilitacji według opisanej metody jest zastosowanie robota, który powtarzałby z pacjentem zadane wzorce odzwierciedlające ruchy wielopłaszczyznowe bez bezpośredniego udziału fizjoterapeuty [11], [12]. Udział fizjoterapeuty sprowadzałby się np. do wprowadzenia wzorca ruchu do pamięci robota połączonego z kończyną rehabilitowanego pacjenta, a następnie ustalenie tempa i ilości jego powtórzeń w treningach biernych. Możliwe byłoby ponadto zastosowanie treningów aktywnych, w których pacjent wykonuje zadany ruch, a robot będzie pozwalał na wykonanie właściwej trajektorii ruchu i dodatkowo będzie aktywnie oporował podczas jego wykonywania. Rozwiązanie to umożliwiłoby jednemu wykwalifikowanemu fizjoterapeucie prowadzenie

jednoczesnej rehabilitacji kilku pacjentów. Dodatkowo, zastosowanie robota w procesie rehabilitacji pozwala na realizację treningu rehabilitacyjnego w formie gier komputerowych, co może podnieść atrakcyjność treningu i zaangażowanie pacjenta w wykonywanych ćwiczeniach [13]. Jednak podczas treningu, np. po udarze mózgu, ważne jest nie tylko wykonywanie samego ruchu rehabilitowaną kończyną, ale także zaangażowanie kory ruchowej mózgu pacjenta tak, aby pod wpływem odpowiednich bodźców zewnętrznych pobudzić proces neuroplastyczności i nauczyć „na nowo” kontrolować kończyny pacjenta. Kluczowym elementem w tym procesie jest biofeedback w postaci zadań ruchowych np. prezentowanych pacjentowi wizualnie. Wykorzystanie robotów w rehabilitacji pozwala odciążyć rehabilitanta od wykonywania pracy fizycznej, a także daje nowe możliwości terapeutyczne. Robot rehabilitacyjny może służyć, za pomocą wbudowanych czujników, jako doskonałe narzędzie diagnostyczne, realizując np. pomiar zakresów ruchomości w stawach, pomiar siły mięśni, czy badanie elektromiograficzne mięśni. Dzięki tak uzyskanym danym, z sesji na sesję rehabilitacyjną możemy pacjentowi przedstawić np. powiększające się zakresy ruchowe w wybranych stawach lub zwiększoną siłę mięśni, co może pozytywnie motywować pacjenta do dalszej pracy. Jest to szczególnie istotne z uwagi na często wielomiesięczny proces rehabilitacji, której poddawany jest pacjent.

W większości przypadków rehabilitacja kończyn jest procesem długotrwałym. Przywrócenie utraconej sprawności kończyny wymaga zazwyczaj wielomiesięcznych ćwiczeń realizowanych przez doświadczonego fizjoterapeutę. Fizjoterapeuta często wykonuje znaczący wysiłek fizyczny, angażując dużą siłę podczas manualnego wykonywania ruchów rehabilitowanych kończyn pacjenta. By ograniczyć zmęczenie rehabilitanta i jednocześnie zapewnić większą powtarzalność aplikowanych ćwiczeń zaczęto wprowadzać urządzenia techniczne, które w późniejszym czasie przybrały formę robotów rehabilitacyjnych. Pierwszym, dobrze udokumentowanym przykładem wspomaganie pracy rehabilitanta przez urządzenia techniczne mogą być mechaniczne urządzenia opracowane przez szwedzkiego lekarza ortopedę Jonasa Gustawa Wilhelma Zandera (1835-1920) [14], który stworzył bogatą kolekcję urządzeń mechanicznych wspomagających proces rehabilitacji, Rys. 1.1.



Rys. 1.1 Przykłady urządzeń rehabilitacyjnych Wilhelma Zandera [14]

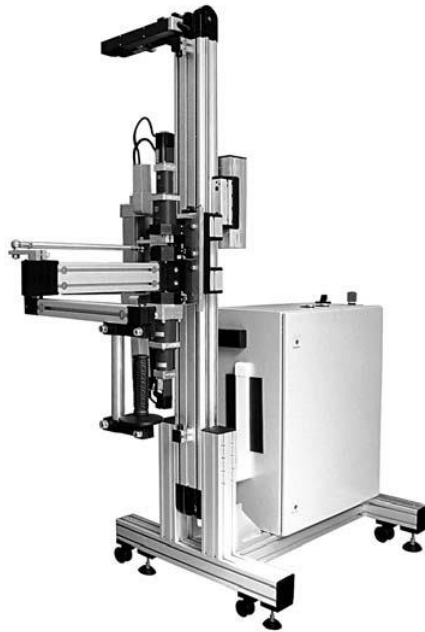
Natomiast, pierwszym robotem rehabilitacyjnym uznanym w literaturze jest MIT-Manus, opracowany przez amerykański uniwersytet Massachusetts Institute of Technology (MIT), nad którym prace rozpoczęto w roku 1989. Robot powstał by ustalić, podczas testów klinicznych, czy powtarzalne ćwiczenia, polegające na ruchu sięgania, mogą przyspieszyć powrót funkcji ramienia u osób po udarze z niedowładem połowicznym.



Rys. 1.2 Robot MIT-Manus (Interactive Motion Technologies, Cambridge, MA) [15]

MIT-Manus jest robotem typu manipulator, który umożliwiał realizację ruchu w płaszczyźnie poziomej (Rys. 1.2). Podczas ćwiczenia, robot mógł wspomagać pacjenta lub stawiać opór a zadaniem pacjenta było, trzymając uchwyt manipulatora, wodzić uchwytem np. w kierunku przemieszczającego się punktu. System komputerowy robota śledził szybkość reakcji, płynność wykonywanego ruchu i przyłożone siły, co dawało rehabilitantowi bardzo ważne informacje odnośnie stanu pacjenta i postępów prowadzonej rehabilitacji. Wspomaganie

prowadzonego ruchu przez ramię robota, jest szczególnie użyteczne na początku procesu rehabilitacji neurologicznej, gdy pacjent dysponuje niewielką siłą mięśni lub rehabilitowana kończyna pozostaje bezwładna [15], [16].



Rys. 1.3 Robot RENUUS-1 (PIAP) [17]

Idea podobna do opisywanego wcześniej robota MIT-Manus została zastosowana w polskim robocie rehabilitacyjnym kończyn górnych RENUUS-1, którego prototyp opracowano w Przemysłowym Instytucie Automatyki i Pomiarów PIAP. Prace nad tym robotem rozpoczęto roku 2005. Podstawowym elementem konstrukcji mechanicznej systemu jest manipulator (Rys. 1.3), który prowadzi ramię pacjenta po określonej wielopłaszczyznowej trajektorii. Ramię manipulatora składa się z dwóch sztywnych elementów połączonych przegubami i mocowanych poprzez przegub do wózka poruszającego się po pionowych prowadnicach przymocowanych do nieruchomej kolumny. Z kinematycznego punktu widzenia konstrukcja mechaniczna urządzenia jest mechanizmem o trzech stopniach swobody. Ramię zakończone jest manipulatorem wyposażonym w czujniki: siły i momentu obrotowego. System robota RENUUS-1 może pracować w czterech trybach: 1 – rejestrowanie trajektorii przez manipulator (tryb uczenia się) poprzez prowadzenie ramienia robota po wymaganej przez terapeutę trajektorii; 2 – trening pasywny (manipulator odtwarza zadaną trajektorię); 3 – trening aktywny (pacjent odtwarza trajektorię wzdłuż modelu wyświetlanego na ekranie); 4 – równoważenie ciężaru ramienia pacjenta umieszczonego bezwładnie w gnieździe stawu robota [17], [18], [19].



Rys. 1.4 Robot ARMin II (ETH Zurich) [20]

Kolejnym interesującym projektem robota rehabilitacyjnego kończyn górnych jest ARMin I, jest to robot o czterech stopniach swobody. Prace nad prototypem tego robota rozpoczęły się w roku 2003 na Politechnice Federalnej w Zurychu (ETH Zürich). Wersja robota ARMin II, nad którą prace rozpoczęto w 2007 roku, miała postać egzoszkieletu o 6 stopniach swobody. ARMin II umożliwia pracę w 3 trybach pracy: trening pasywny, trening aktywny pod postacią gier, trening aktywny pod postacią wykonywania zadań z życia codziennego ADLs (ang.: Activities of Daily Living) np. wykonywania ruchów: podnoszenia szklanki, ustawiania przedmiotów na stole [20].



Rys. 1.5 Robot ARMin III (ETH Zurich) [21]

W trzeciej wersji robota ARMin, powiększono liczbę stopni swobody do 7, dodając możliwość otwierania i zamykania dłoni. Rozbudowano również scenariusze treningowe ADLs o: gotowanie, napełnianie kubka, otwieranie drzwi, utrzymanie higieny osobistej, automat na

napoje i grę na fortepianie. Pośród wielu udoskonań poprawiono również wytrzymałość konstrukcji mechanicznej [21].



Rys. 1.6 Robot Armeo Power (Hocoma) [22]

W roku 2011 Politechnika w Zurychu dokonała transferu technologii do firmy Hocoma, która skomercjalizowała robota ARMin III pod nazwą Armeo Power. Robot posiada 7 stopni swobody, każdy stopień wyposażony jest w napęd i dwa czujniki pozycji kątowej. Dodatkowo uchwyt dłoni zawiera czujnik ciśnienia wykorzystany do pomiaru siły uścisku. Oprogramowanie współpracujące z robotem pozwala na łączenie treningów z grami w przestrzeni wirtualnej i biofeedbackiem. Dodatkowo oprogramowanie umożliwia dopasowanie parametrów ćwiczeń do możliwości pacjenta, posiada także narzędzia do oceny postępów terapii [22]. Obecnie, robot Armeo Power jest uznawany za przykład referencyjny robota rehabilitacyjnego.



Rys. 1.7 Robot ReoGo (Motorika) [23]

Przykładem robota typu manipulator jest ReoGo, który umożliwia wykonywanie ćwiczeń przez wodzenie kończyną górną uchwyconą za dłoń lub dłoń i przedramię. Ważnym elementem systemu jest video feedback, za pomocą którego realizowane są gry lub zadania rehabilitacyjne typu ADLs [23].



Rys. 1.8 Robot HARMONY [24]

W 2015 roku Uniwersytet Teksński w Austin (USA) zaprezentował innowacyjny projekt dwuramiennego robota rehabilitacyjnego typu egzoskielet, będący wynikiem kilkuletnich badań. Przedstawiona konstrukcja posiada w sumie 14 stopni swobody i umożliwia realizację anatomicznych ruchów w szerokich zakresach. Robot korzysta z czujników nacisku np. do podtrzymywania kończyn pacjenta, by nawet słaby pacjent był w stanie wykonać zadany ruch kończynami górnymi [24].



Rys. 1.9 Robot DIEGO (TyroMotion) [25]

Warto również zwrócić uwagę na ciekawe rozwiązanie wykorzystane w robocie Diego, firmy TyroMotion, gdzie do rehabilitowania kończyny wykorzystuje się cięgna podtrzymujące każdą kończynę w dwóch miejscach, umożliwiając wykonywanie ćwiczeń pacjentom w początkowej fazie rehabilitacji w odciążeniu. Robot pozwala na wykonywanie ćwiczeń obiema kończynami równocześnie. System sterowania robota śledzi ruch kończyn w przestrzeni i wykonywanie scenariuszy treningowych w przestrzeni 2D lub 3D [25].



Rys. 1.10 Robot LUNA (EgzoTech) [26]

Bardzo ciekawym przykładem polskiego robota rehabilitacyjnego, który odniósł komercyjny sukces, jest robot Luna, gliwickiej firmy EgzoTech. Jest to robot o tylko jednym stopniu swobody, ale dzięki możliwości pracy w różnych konfiguracjach, przez zmianę końcówki robota, jest bardzo uniwersalny i konkurencyjny cenowo w stosunku do robotów o bardziej skomplikowanych konstrukcjach mechanicznych. Robot Luna umożliwia realizację treningów biernych i czynnych, a także wspomaga ruch kończyny za pomocą elektromiografii. Robot ma wbudowany elektromiograf mogący rejestrować sygnały z maksymalnie 6 kanałów [26], [27].



Rys. 1.11 Robot ARM-100 (ITAM Zabrze) [28]

Trzecim przykładem polskiego robota rehabilitacyjnego, w tym zestawieniu, jest prototyp ARM-100, który jest robotem typu egzozskielet. Prace nad tym robotem rozpoczęły się w Instytucie Techniki i Aparatury Medycznej ITAM w Zabrzu w roku 2007. Jest to robot o 7 stopniach swobody, przeznaczony do realizacji ćwiczeń pasywnych i aktywnych [28], [29]. Robot ARM-100 jest protoplastą przedstawianego w niniejszej pracy doktorskiej robota ARM-200, który stanowi rozwinięcie możliwości wcześniej opracowanej idei sterowania wykorzystującej czujniki siły do programowania trajektorii.

W literaturze, ze względu na swoją konstrukcję mechaniczną, roboty rehabilitacyjne kończyn górnych dzielone są na dwie podstawowe kategorie: roboty typu manipulator i roboty typu egzozskielet. Dodatkowo, niezależnie od typu konstrukcji mechanicznej, roboty rehabilitacyjne dzieli się według obecności funkcji kontroli impedancji mechanicznej w interakcji robota z kończyną pacjenta lub braku takiej funkcjonalności [30].

Roboty typu manipulator charakteryzują się prostą konstrukcją mechaniczną, która ma kontakt z rehabilitowaną kończyną w jednym miejscu, którym jest przeważnie uchwyt dłoni i/lub przedramię. Podczas pracy z robotem typu manipulator, kończyna pacjenta jest wiedziona za pomocą uchwytu lub pacjent wodzi uchwytem robota. W tego typu robocie nie mamy kontroli ruchu kończyny pacjenta w poszczególnych stawach. Natomiast zaletą takiej konstrukcji jest łatwość przestawienia pracy dla kończyny lewej lub prawej i mniejsze skomplikowanie konstrukcji mechanicznej co przekłada się na mniejszy koszt wytworzenia takiego robota [31, s. 12–15]. Przykładem robota typu manipulator może być: MIT-Manus, RENUS-1, ReoGo, a także roboty Luna i Diego.

Drugim rozwiązaniem jest robot typu egzoskielet, który zapewnia wielopunktowy kontakt z kończyną pacjenta, umożliwiając kontrolę ruchu w poszczególnych stawach kończyny pacjenta oraz precyzyjne określenie pozycji statycznej kończyny. Takie rozwiązanie charakteryzuje się dużym skomplikowaniem konstrukcji mechanicznej robota, z uwagi na konieczność zastosowania wielu stopni swobody. W niektórych konstrukcjach, z uwagi na złożoność mechaniczną, roboty projektowane do rehabilitowania np. prawej kończyny nie umożliwiają na przestawienie do rehabilitowania lewej kończyny [31, s. 12–15]. Przykładami robotów rehabilitacyjnych typu egzoskielet mogą być: rodzina robotów ARMin, Armeo Power, ARM-100 i robot Harmony, który jako jedyny w powyższym zestawieniu umożliwia pracę dwukończynową.

Przedstawione urządzenia i inne niewymienione wyżej roboty rehabilitacyjne, umożliwiają prowadzenie różnych scenariuszy treningowych, umożliwiających trening: pasywny, aktywny ze wspomaganie, aktywny z oporowaniem, a także trening dwukończynowy tzw. lustrzany, w których porażona kończyna śledzi ruchy kończyny zdrowej[32]. Pokazuje to różnorodność zastosowanych rozwiązań i możliwości robotów rehabilitacyjnych. Może to być proste urządzenie poruszające kończyną pacjenta za pomocą jednego napędu, takie jak robot Luna lub umożliwiające ruch kończyny w jednej płaszczyźnie, jak w przypadku robota MIT-Manus. Istnieją także rozwiązania, które pozwalają na poruszanie kończyną w przestrzeni z wykorzystaniem siedmiu stopni swobody, czego przykładem jest np. robot Armeo Power. Podczas projektowania robotów, ich projektanci starają się pogodzić ze sobą, czasem sprzeczne wytyczne. Z jednej strony robot powinien być maksymalnie uniwersalny, by umożliwić rehabilitację pacjentów z różnymi dysfunkcjami i na różnych etapach procesu rehabilitacji.

Z drugiej strony, istotne jest, aby koszt urządzenia był na tyle niski, by umożliwić jego zakup przez liczne ośrodki rehabilitacyjnych czy nawet przez osoby prywatne do użytku domowego [33]. Jak widać nie ma robota idealnego, który zdominowałby inne rozwiązania. Zaproponowana w niniejszej pracy doktorskiej uniwersalna koncepcja systemu sterowania robota rehabilitacyjnego umożliwia realizację robota rehabilitacyjnego o jednym lub o wielu stopniach swobody, umożliwiającą wykonywanie treningu w różnych scenariuszach, a do jego budowy wykorzystano tanie i dostępne komponenty.

2. SYSTEMY STEROWANIA SPOTYKANE W ROBOTYCE

Pierwsze roboty, które powstawały już w latach 80-tych XX wieku, były budowane z myślą o zastosowaniach przemysłowych. Przemysł robotyczny opracował kilka standardów, których użycie jest rozpowszechnione w konstrukcjach robotów. Takim przykładem może być wykorzystanie magistrali CAN do komunikacji z czujnikami i elementami wykonawczymi robotów [34].

Jednym z najpopularniejszych rozwiązań szeroko stosowanych w badaniach i rozwijaniu zaawansowanych systemów robotycznych jest platforma programistyczna ROS (Robot Operating System). Jednak nie jest to system operacyjny, tylko oprogramowanie dedykowane dla platform opartych na systemie Linux (zalecana dystrybucja Ubuntu) lub MacOS. System ROS dostarcza zestaw narzędzi, bibliotek i protokołów komunikacyjnych, które umożliwiają programowanie, zarządzanie i kontrolę pracy robotów. ROS jest modułowym i rozszerzalnym systemem, który głównie skupia się na obsłudze języka C++, Python i Lisp, ale również eksperymentalnie obsługuje inne języki jak Java czy Lua. Jest to bardzo uniwersalne narzędzie współpracujące z dodatkowymi bibliotekami np. Gazebo do wizualizacji i symulacji robotów, co pozwala realizować skomplikowane scenariusze eksperymentów [35].

W przeciwieństwie do otwartej platformy ROS, każdy z głównych producentów robotów przemysłowych oferuje własne komercyjne systemy sterowania robotami, takie jak np.:

- Fanuc HandlingTool Software
- ABB RobotStudio
- Siemens ROBOTSTAR
- Yaskawa MotoPlus/MotoLogix
- KUKA.KRC

Ważnym pytaniem, na początku prac nad systemem sterowania robota, było czy do komunikacji z modułami robota zastosować magistralę CAN, która posiada wiele zalet takich jak:

- wysoka niezawodność,
- możliwość zastosowania topologii multi-master,
- mechanizm arbitrażu rozszczekujący, który Master może w danej chwili nadawać,
- możliwość transmisji typu broadcast (do wszystkich węzłów sieci),
- mechanizm retransmisji zakłóconych wiadomości,
- wysoka szybkość transmisji do 1Mbit/s.

Natomiast jedną z istotnych wad magistrali CAN, w przypadku projektowanego systemu, było ograniczenie liczby przesyłanych danych w jednym datagramie do maksymalnie 8 bajtów. W planowanym scenariuszu pracy chcemy, by w jednym datagramie była możliwość przesłania kompletu informacji wymaganych do sterowania i zwrócenie kompletu informacji do oceny stanu danego węzła. W opisanym niżej rozwiązaniu, datagramy na magistrali wewnętrznej, mają stałą wielkość 48 bajtów, gdzie 40 bajtów jest przeznaczone na przesyłanie parametrów pracy węzła. Takie podejście znacznie upraszcza i przyspiesza sterowanie pracą systemu. Dodatkowym ułatwieniem jest fakt, że w projektowanym rozwiązaniu, wymagany jest tylko jeden węzeł Master, który odpytuje po kolei węzły Slave. Opracowane rozwiązanie zostało oparte na magistrali RS-485, która cechuje się dużą odpornością na zakłócenia przesyłanych informacji z uwagi na wykorzystanie do transmisji danych skręconej pary przewodów. Zapewniona jest również duża szybkość transmisji danych oraz wykorzystanie tylko 2 przewodów do transmisji w trybie half-duplex, ponieważ magistrala przełącza kierunek transmisji danych. Oznacza to, że nie jest możliwe jednoczesne nadawanie i odbieranie danych, jednak w prezentowanym rozwiązaniu taka funkcjonalność nie jest wymagana.

Podczas analizy wymagań dla projektowanego systemu sterowania robota rehabilitacyjnego stwierdzono, że do realizacji zaplanowanych zadań nie jest konieczne wdrożenie rozwiązania bardzo uniwersalnego. Natomiast ważnym cechę wybranego rozwiązania powinna być łatwa integracja z oprogramowaniem .NET, z którym docelowo będzie współpracować robot, którego system sterowania będzie opisywany w niniejszej pracy.

3. CEL I ZAKRES PRACY

Prace badawcze prowadzone w Instytucie Techniki i Aparatury Medycznej ITAM w latach 2007-2009 zaowocowały opracowaniem prototypu robota rehabilitacyjnego ARM-100. Robot ten został zaprojektowany w formie egzoszkieletu, który umożliwia realizację wielopłaszczyznowych ruchów terapeutycznych we wszystkich stawach kończyny górnej człowieka, a system sterowania opracował autor niniejszej pracy. Po zakończeniu testowania prototypu robota rehabilitacyjnego ARM-100 określono kierunki dalszego rozwoju opracowanego rozwiązania. Jednym z głównych postulatów było wskazanie konieczności opracowania nowego, rozbudowanego systemu sterowania umożliwiającego pracę robota w tzw. trybie „śledzenie ruchu”, czyli takiego systemu sterowania, który na podstawie danych z czujników pomiarowych potrafiłby automatycznie określać intencję ruchu w stawach pacjenta, a elementy napędowe robota wspomagałyby ten ruch.

W roku 2017 w ITAM (obecnie Sieć Badawcza Łukasiewicz – Krakowski Instytut Technologiczny) rozpoczęto prace mające na celu opracowanie nowej wersji robota rehabilitacyjnego kończyn górnych. Realizacja niniejszego doktoratu wdrożeniowego była ściśle powiązana z tymi pracami, a główny cel rozprawy doktorskiej sformułowano w następujący sposób:

Celem doktoratu wdrożeniowego było opracowanie modułarnego systemu sterowania robota rehabilitacyjnego przeznaczonego do terapii kończyn górnych wykorzystującego sprzężenie siłowe do realizacji ruchów robota.

Na podstawie przyjętych założeń i głównego celu sformułowano tezy niniejszej rozprawy:

1. Dane z czujników pomiarowych siły umieszczonych w poszczególnych węzłach robota rehabilitacyjnego można wykorzystać do automatycznej detekcji intencji ruchu w stawach kończyny górnej pacjenta oraz do sterowania trajektorią ruchu robota.
2. Wprowadzenie w algorytmie sterowania ruchami robota minimalnej siły wymaganej do aktywacji napędów przyczyni się do zwiększenia bezpieczeństwa pacjenta poprzez wyeliminowanie niepożądanych ruchów robota.
3. Wykorzystanie aplikacji sterującej działającej w systemie operacyjnym nie czasu rzeczywistego (ang.: non-real-time system), do kontroli pracy robota rehabilitacyjnego, nie wpłynie negatywnie na pracę systemu sterowania robota.

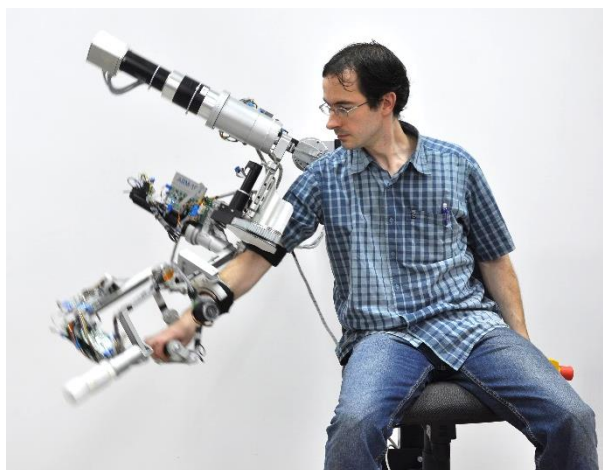
4. OPIS SYSTEMU STEROWANIA ROBOTA REHABILITACYJNEGO

Przed przystąpieniem do prac projektowych nad systemem sterowania robota rehabilitacyjnego należało ustalić oczekiwane funkcjonalności, które powinien przysłużyć system posiadać. Pierwszym krokiem w realizacji tego zadania był przegląd rozwiązań medycznych robotów rehabilitacyjnych, które są już dostępne na rynku lub są dopiero w fazie prototypu, w tym prace własne nad prototypem robota ARM-100. Prototyp ten był punktem wyjścia do opracowania kolejnej generacji robotów rehabilitacyjnych opracowywanych w ramach prac prowadzonych w Sieci Badawczej Łukasiewicz – Krakowskim Instytucie Technologicznym, a wcześniej w Instytucie Techniki i Aparatury Medycznej w Zabrze.

Na podstawie przyjętych koncepcji i założeń do wymaganych funkcji systemu sterowania robota rehabilitacyjnego kończyn górnych, można było przejść do realizacji drugiego kroku przygotowawczego, jakim była analiza wymagań prawnych i normatywnych.

Ważnym wstępem do analizy wymagań funkcjonalnych były doświadczenia zdobyte podczas prac poprzedzających projekt doktorski, były to prace nad prototypem robota rehabilitacyjnego kończyny górnej ARM-100, którego system sterowania projektował autor niniejszej pracy. Urząd Patentowy RP objął ochroną patentową opracowane rozwiązanie pod nazwą „Układ sterowania napędami egzoszkieletu”, Pat.218184. W ramach prac została opracowana koncepcja sterowania i podstawowe elementy systemu sterowania robotem, w postaci modułu Master i modułu Slave, który pozwalał na realizację wybranego trybu pracy i obsługę czujników węzła robota [36].

Opracowany robot ARM-100 pozwolił zweryfikować przyjętą koncepcję sterowania opartą na czujnikach siły a także możliwość rozproszenia elektroniki pomiarowo-wykonawczej na konstrukcji mechanicznej robota o siedmiu stopniach swobody, przedstawionego na **Błąd!**
Nie można odnaleźć źródła odwołania..



Rys. 4.1 Zdjęcie robota ARM-100

Prototyp opracowanego robota ARM-100 poddano badaniom testującym, w tym wstępnej weryfikacji doświadczalnej w warunkach laboratoryjnych z uwzględnieniem specyfiki prowadzenia zabiegów rehabilitacyjnych. Wstępna weryfikacja obejmowała ocenę konstrukcji urządzenia oraz systemu sterowania w tym m.in. napędów łańcucha kinematycznego, sztywności konstrukcji oraz sposobu zamocowania czujników siły i skuteczności uzyskiwania z nich informacji. W sposób szczególny skoncentrowano się na analizie kinematycznej urządzenia. Wyniki badań kinematyki prototypu robota rehabilitacyjnego ARM-100 zaprezentowano m.in. w pracy autorstwa A. Michnik et al. [37]. Doświadczenia uzyskane podczas prac związanych z opracowaniem prototypu robota rehabilitacyjnego ARM-100 pozwoliły na opracowanie wymagań funkcjonalnych, które powinny być uwzględnione w systemie sterowania w robocie rehabilitacyjnym ARM-200. Do głównych wymagań funkcjonalnych można zaliczyć:

- możliwość wykonywania ruchów złożonych kończyną górną,
- możliwość wykonywania izolowanych ruchów w poszczególnych stawach,
- możliwość odtwarzania zapamiętanych trajektorii ruchowych,
- możliwość śledzenia ruchu kończyny przez ramię robota z możliwością zadawania progu wartości siły, po którym jest możliwy ruch robota.

Konieczne okazało się również wprowadzenie nowych funkcjonalności (nie uwzględnionych w prototypie robota rehabilitacyjnego ARM-100), takich jak:

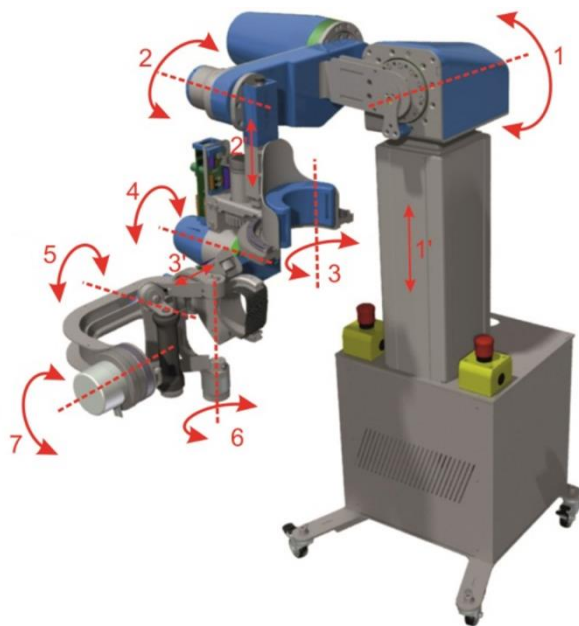
- możliwość zmiany konfiguracji robota dla lewej lub prawej strony,
- możliwość sterowania hamulcami z detekcją uszkodzenia,

- powiększenie mocy obliczeniowej mikrokontrolera sterującego węzłami robota,
- możliwość sterowania siłownikiem rozsuwu w węźle robota,
- zapewnienie automatycznego adresowania węzłów robota,
- możliwość komunikacji robota z oprogramowaniem sterującym za pomocą interfejsu LAN,
- możliwość współpracy węzła robota z wejściówką elektromiograficzną,
- zmiana sposobu łączenia kabli magistrali sterującej, która ułatwia przyszły serwis robota, przez możliwość wymiany pojedynczego odcinka łączącego wybrane węzły,

możliwość ograniczania zakresu ruchu w danym węźle na podstawie aktualnej pozycji innego węzła. To pozwala ograniczyć lub wyłączyć rotację w stawie łokciowym, gdy staw łokciowy jest wyprostowany. Taki mechanizm jest istotny podczas realizacji trybu śledzenia ruchu.

4.1. Prototyp robota rehabilitacyjnego ARM-200

Prototyp robota rehabilitacyjnego kończyn górnych ARM-200, składa się z podstawy i ramienia o konstrukcji egzoszkieletu, posiadającego siedem podstawowych stopni swobody oraz trzy pomocnicze. Szczegółowa analiza kinematyki robota rehabilitacyjnego kończyn górnych ARM-200 została opisana w artykule autorstwa M. Sobiech et al. [38]. Numery par kinematycznych z zaznaczonymi stopniami swobody zostały przedstawione na Rys. 4.2. Podstawowe stopnie swobody tworzą aktywne pary kinematyczne, które umożliwiają wykonanie ruchów kończyną górną, zarówno prawą i lewą. Pomocnicze stopnie swobody służą natomiast możliwości dopasowania egzoszkieletu do pacjenta, poprzez regulację wysokości ramienia względem podłoża oraz długości członu ramienia i przedramienia robota. Przedstawiony poniżej schemat kinematyczny robota ARM-200 pozwala na wykonywanie ruchów w następujących stopniach swobody:



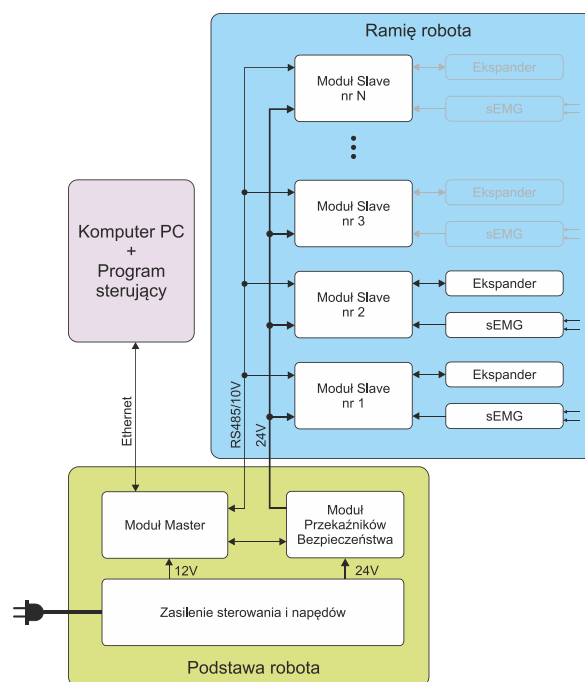
1. Staw barkowy:
odwiedzenie – przywiedzenie.
 2. Staw barkowy:
zgięcie – przeprost.
 3. Staw barkowy:
rotacja wewnętrzna – zewnętrzna.
 4. Staw łokciowy:
wyprost – zgięcie.
 5. Staw nadgarstkowy:
zgięcie promieniowe – łokciowe.
 6. Staw nadgarstkowy:
zgięcie dłoniowe – grzbietowe.
 7. Staw łokciowy:
supinacja – pronacja.
- 1'. Siłownik zmiany wysokości
2'. Siłownik długości ramienia
3'. Siłownik długości przedramienia

Rys. 4.2 Wizualizacja par kinematycznych robota ARM-200 z zaznaczonymi stopniami swobody (1-7 podstawowe stopnie swobody, 1'-3' pomocnicze stopnie swobody).

Rolą systemu sterowania robota rehabilitacyjnego jest zapewnienie dwukierunkowej komunikacji pomiędzy aplikacją sterującą uruchomioną na komputerze PC a ramieniem robota, który pełni rolę elementu wykonawczo-pomiarowego. Istotnymi parametrami systemu sterowania jest jego niezawodność i responsywność. Dodatkowo, z uwagi na specyfikę konstrukcji mechanicznej robota polegającą na zastosowaniu przewodów łączących elementy

systemu i możliwość uszkodzeń tych przewodów, system sterowania powinien posiadać mechanizmy wykrywania i sygnalizacji uszkodzeń, a także, co jest bardzo istotne, powinien zapewniać przejście do stanu bezpiecznego w przypadku wystąpienia awarii.

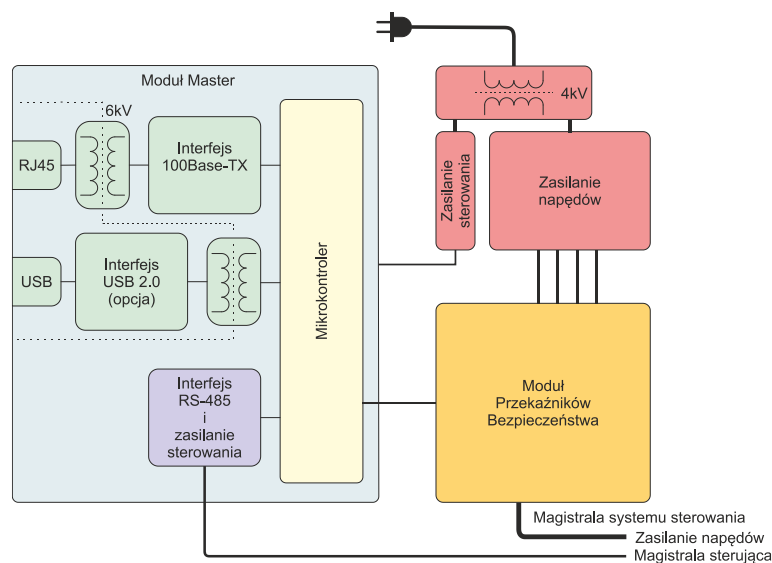
System sterowania robota rehabilitacyjnego składa się z trzech głównych składników przedstawionych na Rys. 4.3. Integralną częścią systemu sterowania robotem jest komputer PC, na którym zainstalowane jest oprogramowanie sterujące procesem rehabilitacji. Komputer PC łączy się z modułem Master, który jest pośrednikiem pomiędzy komputerem PC, a węzłami Slave systemu sterowania. Każdy moduł Slave realizuje zadany tryb pracy na podstawie elementów pomiarowych i umożliwia kontrolę pracy jednego silnika typu BLDC, czyli silnika prądu stałego z komutacją elektroniczną. Dodatkowo, każdy z modułów Slave umożliwia współpracę z modułem Expandera, który rozszerza funkcjonalność węzła o dodatkowe funkcje, takie jak np. sterowanie hamulcem silnika czy sterowanie siłownikiem liniowym rozsuwu. W obecnej postaci robota, moduł Expandera jest wymagany tylko w trzech węzłach systemu. Kolejnym dodatkowym modułem, o który może być rozszerzony dany węzeł robota jest 2-kanałowy moduł sEMG (ang.: surface electromyography), mierzący aktywność elektryczną mięśni za pomocą naklepanych elektrod. Opracowany system sterowania został zaprojektowany jako rozwiązanie uniwersalne, pozwalające na pracę w robotach o dowolnej liczbie stopni swobody. Konfiguracja sprzętowa wraz z adresowaniem węzłów systemu jest przeprowadzana automatycznie.



Rys. 4.3 Schemat systemu sterowania robota ARM-200.

4.2. Moduł Master

Pierwszym elementem sprzętowym robota jest moduł Master, który jest pośrednikiem pomiędzy interfejsem Ethernet 100Mb a interfejsem szeregowym RS-485. Dodatkową rolą modułu Master, przedstawionego na Rys. 4.4, jest kontrola modułu przekaźników bezpieczeństwa, które umożliwiają zatrzymanie napędów robota w dowolnej chwili za pomocą wyłączników awaryjnych lub przez moduł Master w przypadku wykrycia awarii. Warto też zwrócić uwagę na transformator separacyjny zastosowany na wejściu zasilania ~230V, który zapewnia separację galwaniczną 4kV. Dodatkowo, zastosowany zasilacz sterowania i zasilacze napędów spełniają normę medyczną IEC60601-1, co oznacza również wytrzymałość elektryczną pomiędzy wejściem i wyjściem 4kV. Zapewnienie podwójnej bariery galwanicznej 4kV jest wymogiem normy na urządzenia medyczne [39]. Również interfejs Ethernet standardowo wyposażony jest w transformator separujący galwanicznie, ale tylko do poziomu 1,5kV. Dlatego, w module Master zastosowano transformator o podwyższonej wytrzymałości elektrycznej do 6kV.



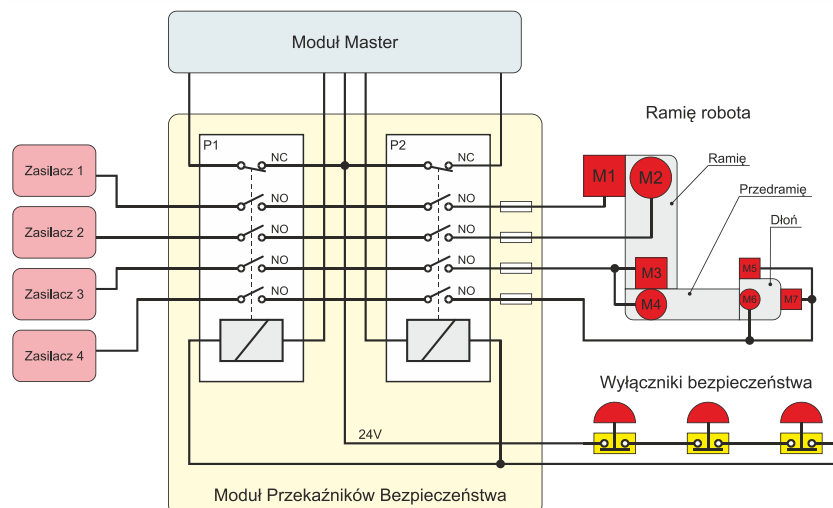
Rys. 4.4 Schemat blokowy modułu Master wraz z zasilaniem i modulem przekaźników bezpieczeństwa.

Głównym elementem modułu Master, jest 32-bitowy mikrokontroler z rdzeniem ARM Cortex M4, który taktowany jest zegarem 240MHz. Wybrany mikrokontroler zapewnia odpowiedni poziom mocy przetwarzania, a także wyposażony jest w interfejs Ethernet umożliwiający realizację komunikacji z komputerem sterującym. Poza podstawowymi

funkcjami komunikacyjnymi, moduł Master wyposażony jest w cztery kanały kontroli siłowników, umożliwiające kontrolę pozycji fotela pacjenta.

4.3. Moduł Przełączników Bezpieczeństwa

Moduł Master współpracuje z Modułem Przełączników Bezpieczeństwa, którego budowa i elementy towarzyszące zostały przedstawione na Rys. 4.5. Moduł ten, odcina napięcie zasilania napędów w niezależnych 4 kanałach. Napędy stawu barkowego, o największych mocach, zasilane są z niezależnych zasilaczy, kolejne napędy o mniejszych mocach, w stawie łokciowym i napędy dłoni zasilane są przez zasilacze współdzielone. Dodatkowo, każdy z kanałów modułu został zaopatrzony w bezpiecznik topikowy. Moduł Przełączników Bezpieczeństwa umożliwia podłączenie trzech bistabilnych wyłączników bezpieczeństwa, które podczas normalnej pracy mają styki zwarte, rozwarcie styku jednego wyłącznika aktywuje odłączenie zasilania, co prowadzi do włączenia hamulców w napędach, które zatrzymują ruch ramienia robota. Stan odłączenia zasilania jest wykrywany i zatraskiwany przez moduł Master i wymaga intencjonalnego wznowienia pracy robota w aplikacji sterującej systemem.



Rys. 4.5 Uproszczony schemat Modułu Przełączników Bezpieczeństwa i elementów towarzyszących

Do realizacji wyłącznika bezpieczeństwa, zostały użyte specjalne przekaźniki bezpieczeństwa, spełniające normę EN 61810-3, której jednym z wymogów jest wyposażenie przekaźnika w dodatkowy styk pozwalający wykryć tzw. sklejenie styków przekaźnika. Przed włączeniem

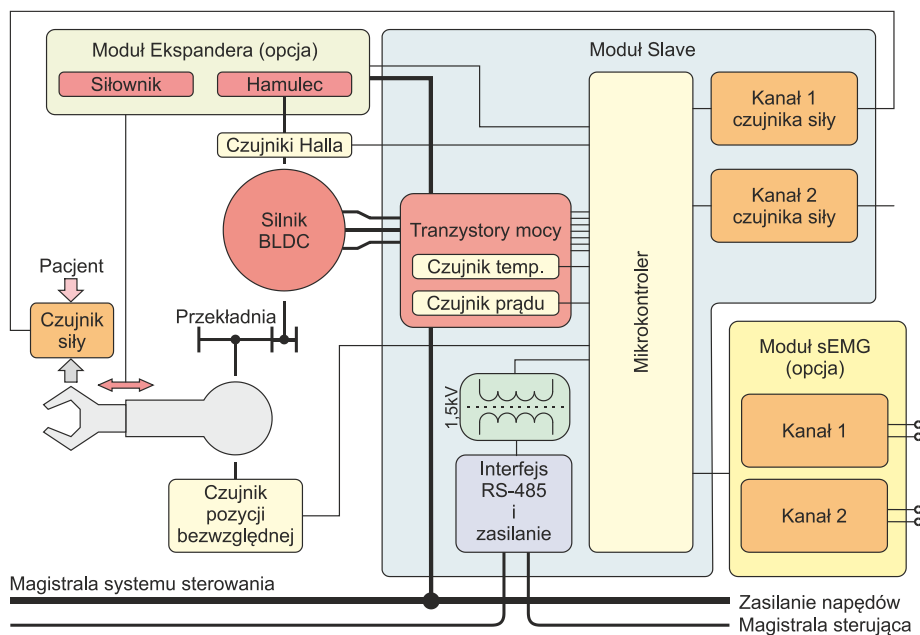
przełącznika wykonywana jest procedura testowa polegająca na sprawdzeniu czy w danym przełączniku nie wystąpiło sklejenie styków. Następnie, po zasileniu cewki przełącznika, sprawdzany jest styk kontrolny czy nastąpiła prawidłowa reakcja. By zwiększyć bezpieczeństwo, zastosowano zdublowanie przełączników bezpieczeństwa. Takie rozwiązanie pozwala odłączyć zasilanie nawet w przypadku uszkodzenia jednego z przełączników. Kontrolę stanu przełączników realizuje oprogramowanie mikrokontrolera modułu Master.

4.4. Moduł Slave

Elementami wykonawczymi systemu sterowania robota są moduły Slave, które kontrolują prace napędów robota na podstawie otrzymanych rozkazów i parametrów odczytanych z czujników współpracujących z danym modułem Slave. Dodatkową informację moduły Slave otrzymują „podszuchując” komunikację modułu Master z innymi modułami Slave. Taki „podszuch” umożliwia realizację dodatkowych funkcji jak np. ograniczanie zakresu ruchu rotacji w stawie barkowym na podstawie wartości kąta zgięcia kończyny w stawie łokciowym. Praca modułu Slave kontrolowana jest za pomocą 32-bitowego mikrokontrolera z rdzeniem ARM Cortex M4, taktowanego zegarem 240MHz. W celu ograniczenia miejsca zajmowanego przez pakiet elektroniki, w przypadku kontroli napędów mniejszej mocy, została opracowana wersja „krótsza” modułu Slave, która współpracuje z napędami o mniejszej mocy.

Schemat blokowy modułu Slave i komponentów z nim współpracujących został zaprezentowany na Rys. 4.6. Jak widać na rysunku, do modułu Slave podłączona jest magistrala sterująca i magistrala mocy do zasilania silnika BLDC. Magistrala sterująca zawiera w sobie również zasilanie małej mocy do zasilania mikrokontrolera i elementów pomiarowych. Takie rozdzielenie zasilania pozwala na odłączenie zasilania mocy nie tracąc informacji z czujników. W celu uzyskania wysokiej odporności komunikacji modułów na zakłócenia, magistrala sterująca systemu i zasilanie sterowania separowane jest galwanicznie od reszty układu barierą o wytrzymałości 1,5kV. Silnik BLCD połączony jest mechanicznie z przekładnią, która porusza elementem wykonawczym robota. Pozycja kątowa elementu wykonawczego robota kontrolowana jest przez dwa czujniki pozycji: bezwzględnej i względnej. Czujnik pozycji bezwzględnej umieszczony jest na elemencie wykonawczym, umożliwia odczytanie pozycji danego elementu mechanicznego po włączeniu zasilania.

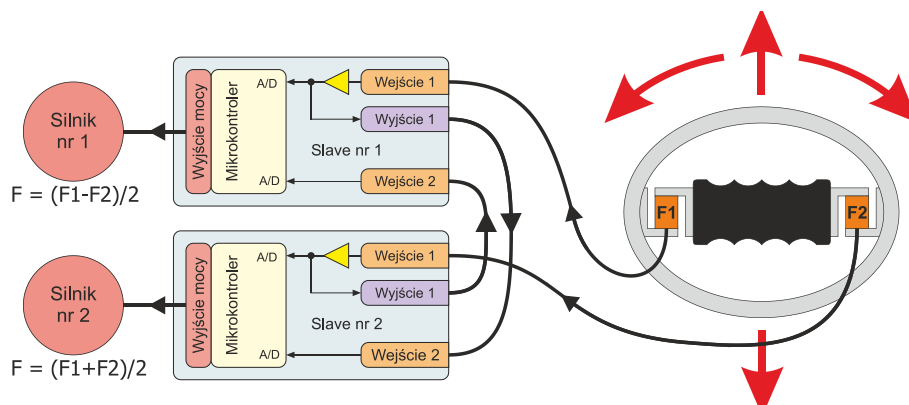
Podczas włączania robota, informacja z czujnika bezwzględnego jest wykorzystana do aktualizacji czujnika względnego wykorzystującego czujniki Halla wbudowane w silnik BLDC. Algorytmy stabilizacji prędkości i pozycji opierają swoją pracę na czujniku względnym. Warto zauważyć, że podstawową rolą czujników Halla wbudowanych w silnik, jest sterowanie procesem komutacji faz za pomocą tranzystorów mocy MOS-FET. Czujnik pozycji bezwzględnej może służyć jako dodatkowe zabezpieczenie zakresu ruchu elementu wykonawczego robota.



Rys. 4.6 Schemat blokowy modułu Slave systemu sterowania robotem.

Poza informacją o pozycji kątowej, bardzo ważną informacją jest siła występująca pomiędzy elementem wykonawczym robota, a kończyną pacjenta. Założeniem przedstawianego rozwiązania jest, by każdy kontakt człowiek-robot był wyposażony w czujnik siły. Jak widać na Rys. 4.6, Moduł Slave wyposażony jest w dwa kanały pomiarowe siły. Drugi kanał pomiarowy siły umożliwia podłączenie wzmocnionego analogowego sygnału siły z innego modułu Slave i wzajemnie, pomiędzy dwoma modułami Slave, przekazywanie sobie informacji o mierzonych siłach. Pozwala to na realizację scenariuszy pomiaru sił różnicowych lub wspólnych. Jak widać na Rys. 5.7, wyliczona różnicowa wartość siły może kontrolować napęd rotacji uchwytu dłoni, natomiast wyliczona siła sumacyjna będzie kontrolować zginanie

w stawie nadgarstkowym. Przedstawione rozwiązanie zostało zgłoszone do Urzędu Patentowego RP i jest chronione dwoma patentami PL nr 218185 [40] i nr 218284 [41].



Rys. 4.7 Schemat połączenia w parę dwóch modułów Slave

4.5. Moduł Ekspandera

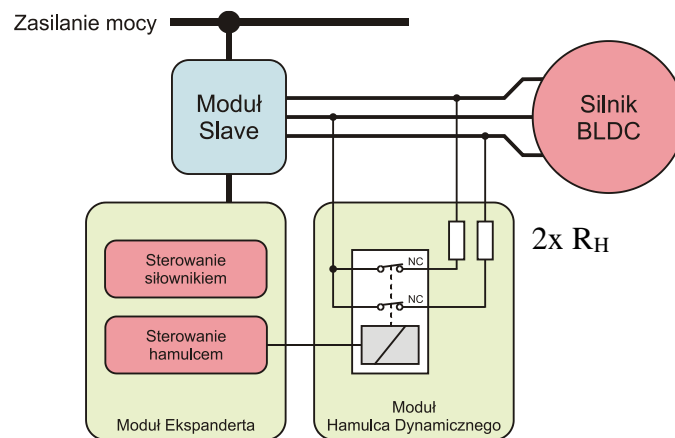
Ważnym elementem robota, ze względu na bezpieczeństwo pacjenta, jest opcjonalny moduł ekspandera. Ten moduł umożliwia rozszerzenie funkcjonalności modułu Slave o możliwość kontroli elektromagnesu hamulca silnika a także możliwość współpracy z siłownikiem rozsuwu wraz z enkoderem rozsuwu. W robocie ARM-200 moduł Ekspandera umożliwia sterowanie zmianą długości części przedramiennej i ramiennej robota a także kontrolę kolumny ustalającej wysokość robota. Siłowniki rozsuwu umożliwiają dopasowanie robota do budowy anatomicznej pacjenta. Takie rozwiązanie pozwala automatycznie i szybko dopasować wymiary robota do kolejnego pacjenta na podstawie parametrów zapamiętanych podczas poprzedniej sesji treningowej. Praca hamulca jest programowo kontrolowana przez moduł Slave.

By wykryć uszkodzenie uzwojenia hamulca lub brak podłączenia, moduł Ekspandera został wyposażony w detektor prądu. Ze względów bezpieczeństwa, hamulce elektromagnetyczne są umieszczone w trzech węzłach robota ARM-200, w których istnieje ryzyko opadnięcia elementu wykonawczego robota pod wpływem grawitacji przy utracie zasilania lub po wyłączeniu programowym silników. Dwa moduły Ekspanderów umieszczone są w węzłach stawu barkowego (przeprost-zgięcie i odwiedzenie) i w węzle stawu łokciowego odpowiedzialnego za zginanie.

4.6. Moduł Hamulca Dynamicznego

Dodatkowo w stawie barkowym węzeł odpowiedzialny za ruch rotacji wewnętrznej-zewnętrznej został wyposażony w hamulec dynamiczny. Hamulec ten znacząco ogranicza szybkość opadania części przedramiennej i jednocześnie umożliwia wykonanie ruchu tą częścią ramienia robota, co pozwala na wyswobodzenie pacjenta w przypadku braku zasilania napędu robota.

Budowę hamulca dynamicznego i schematyczny sposób współpracy z innymi elementami węzła ilustruje Rys. 4.8.



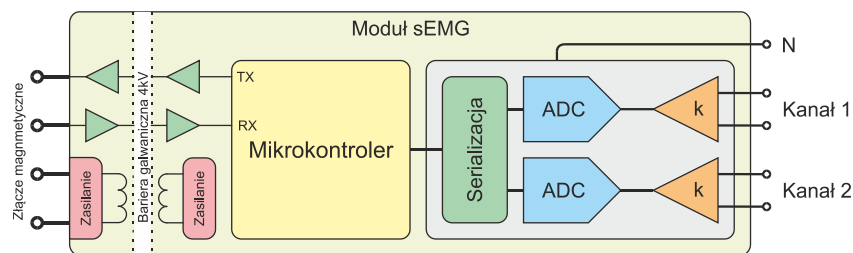
Rys. 4.8 Schemat pracy modułu Hamulca Dynamicznego

Działanie hamulca dynamicznego polega na zwarciu uzwojeń silnika BLCD za pomocą styków przełącznika w przypadku wysterowania przez moduł Slave lub zaniku zasilania. Z punktu widzenia sterowania, moduł Dynamicznego Hamownia zachowuje się jak zwykły hamulec elektromagnetyczny. Podczas normalnej pracy zasilona jest cewka przełącznika i styki są rozwarte. Natomiast, podczas intencjonalnego włączenia hamowania lub zaniku zasilania następuje zanik prądu w cewce przełącznika i zwarcie styków przełącznika. W trakcie opadania części mechanicznej przedramienia, pod wpływem grawitacji, silnik poprzez przekładnię jest rozpędzany. W takim przypadku, silnik BLCD pracuje w trybie prądnicowym, im wyższa prędkość tym powstaje większa siła elektromotoryczna w uzwojeniach silnika, a z uwagi na zwarcie uzwojeń powstaje prąd, który wytwarza moment hamujący. Siłę hamowania można modyfikować rezystancją rezystorów R_H , które ograniczają prąd zwarcia. Rezystory R_H pełnią

również funkcję zabezpieczenia na wypadek wystąpienia stanu przejściowego po zaniku zasilania, gdy jeszcze pracują tranzystory mocy modułu Slave.

4.7. Moduł sEMG

Każdy z modułów Slave może współpracować z modułem sEMG, za pomocą dedykowanego złącza magnetycznego, którego styki umożliwią przekazanie zasilania a także transmisję danych przez interfejs szeregowy UART. Z uwagi na część aplikacyjną interfejsu podłączaną elektrycznie do skóry pacjenta, interfejs został wyposażony w dodatkową barierę galwaniczną o wytrzymałości napięciowej 4kV, która została przedstawiona na schemacie blokowym modułu sEMG (Rys. 4.9). Obecnie, informacja o pobudzeniu elektrycznym mięśni z modułu sEMG, przekazywana jest w postaci uśrednionej wartości z obwiedni sygnału, niezależnie z dwóch kanałów, podobnie jak parametr mierzonej siły. Moduły są dołączane za pomocą magnetycznych zaczepek. Do pomiaru elektrycznej aktywności mięśni został wykorzystany specjalizowany układ ADS1292R[42], który w swojej strukturze zawiera dwa 24-bitowe przetworniki sigma-delta, przedwzmacniacze z programowanym wzmocnieniem i układy detekcji braku kontaktu elektrod. Kontrola pracy modułu sEMG została zrealizowana za pomocą mikrokontrolera STM32F051C8[43], który posiada rdzeń ARM Cortex-M0.



Rys. 4.9 Schemat blokowy modułu sEMG

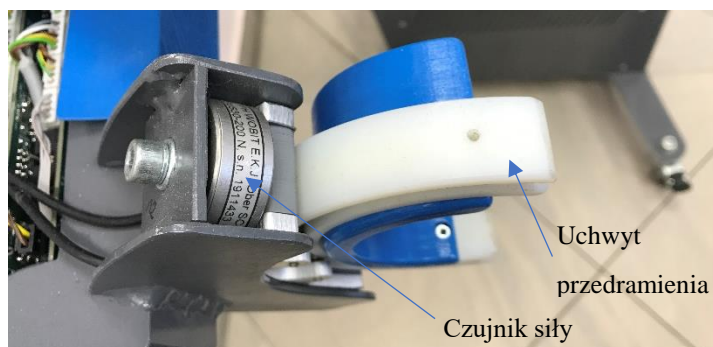
Wykonany moduł sEMG, wraz z przewodami, został przedstawiony na Rys. 4.10. Moduł współpracuje ze standardowymi elektrodami samoprzylepnymi do badań EMG/EKG. Do podłączenia kanału różnicowego wymagane są dwie elektrody i dodatkowo jedna elektroda neutralna (N) dla dwóch kanałów.



Rys. 4.10 Zdjęcie modułu sEMG wraz z przewodami

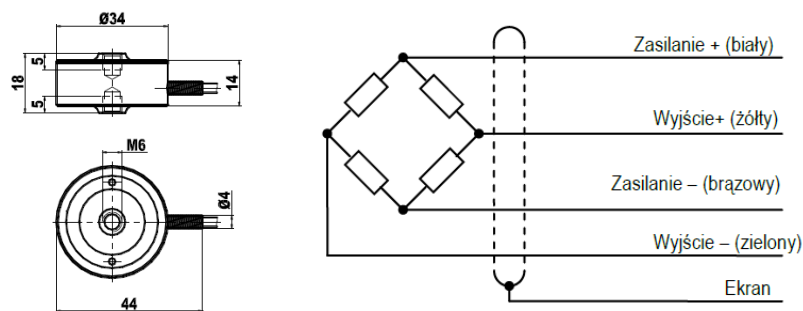
4.8. Czujnik siły

Istotnym elementem pomiarowym systemu sterowania są czujniki siły. Każdy z uchwytów kończyny, łączących kończynę pacjenta z ramieniem robota, wyposażony jest w czujnik lub czujniki siły. Na Rys. 4.11 przedstawiony jest uchwyt przedramienia pacjenta, który wyposażony jest w dwa czujniki siły za pomocą, których możliwy jest pomiar nacisku w dwóch kierunkach. Za pomocą czujników siły możliwa jest rejestracja ruchu lub realizacja treningu aktywnego, w którym ruch jest możliwy dopiero po przekroczeniu przez pacjenta zadanej wartości siły. W treningu biernym, w którym ruch zadawany jest przez napędy robota, czujniki siły służą do zabezpieczania pacjenta przed kontuzją. Wykrycie zbyt dużej siły podczas treningu biernego może sygnalizować wystąpienie skurczu spastycznego mięśni rehabilitowanej kończyny. Tor pomiarowy siły modułu Slave został zrealizowany za pomocą instrumentalnego wzmacniacza INA333[38], który charakteryzuje się dużą precyzją pomiaru napięcia i odpornością na zmiany temperatury.



Rys. 4.11 Zdjęcie przedstawiające czujnik siły w uchwycie przedramienia robota ARM-200

W opisywanym projekcie wykorzystano miniaturowe czujniki sił z rodziny EMS30 firmy EMSYST w dwóch zakresach pomiarowych: 100N i 200N [44]. Czujnik siły jest belką tensometryczną, składającego się z czterech tensometrów naklejonych na odkształcane elementy i połączonych w układzie mostka Wheatstone’a [45, s. 25–28]. Zastosowanie mostka Wheatstone’a pozwala zwiększyć czułość układu pomiarowego, a także skompensuje negatywny wpływ zmiany temperatury i zmian napięcia zasilania. Budowa zastosowanego czujnika siły wraz ze schematem połączeń został przedstawiony na Rys. 4.12.



Rys. 4.12 Rysunek czujnika siły EMS30 wraz ze schematem połączeń [44]

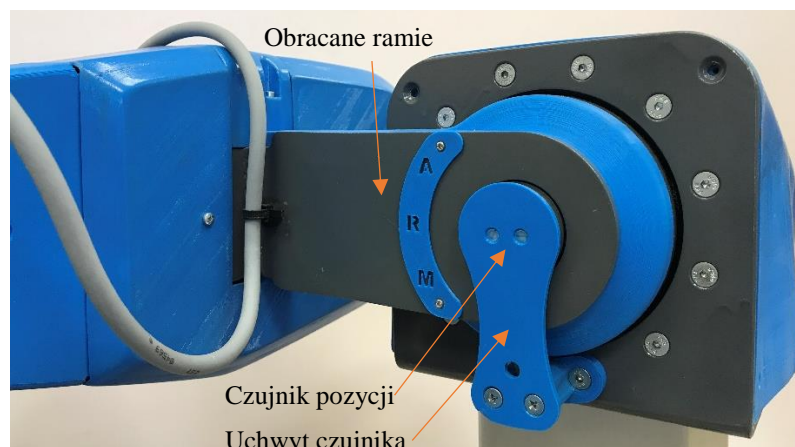
4.9. Czujnik pozycji bezwzględnej

W celu realizacji funkcji pomiaru pozycji kątowej każdej z osi swobody robota, został opracowany bezwzględny enkoder pozycji kątowej na bazie 12-bitowego czujnika magnetycznego AS5600 firmy AMS, który przedstawiono na Rys. 4.13.



Rys. 4.13 Zdjęcia bezwzględnego enkodera pozycji kątowej

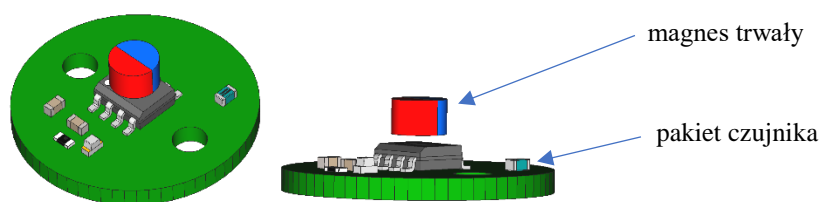
Każda z osi swobody robota wyposażona jest w czujnik bezwzględnej pozycji kątowej, który umieszczony jest na elemencie wykonawczym, za przekładnią danego napędu, co pozwala odczytać pozycję wszystkich elementów ramienia robota po włączeniu zasilania. Przykład wykorzystania czujnika został przedstawiony na Rys. 4.14.



Rys. 4.14 Zdjęcie zabudowanego czujnika bezwzględnej pozycji kątowej węzle nr 1 robota ARM-200

Odczyt pozycji bezwzględnej następuje po włączeniu zasilania i jest przepisywany do licznika pozycji względnej w module Master, który później aktualizowany jest na podstawie impulsów z czujników Halla wbudowanych w silnik BLDC, sterujących również procesem komutacji tranzystorów mocy. Podczas normalnej pracy stabilizacja pozycji kątowej realizowana jest na podstawie licznika pozycji względnej, a cyklicznie (co 50ms) odczytywana pozycja bezwzględna służy jedynie do kontroli pozycji kątowej danej osi.

Czujnik pozycji kątowej współpracuje z dedykowanym magnesem o polaryzacji wzdłuż średnicy. Wbudowany element pomiarowy w postaci mostka składającego się z czterech czujników Halla, odczytuje pozycje magnesu umieszczonego nad elementem pomiarowym, co zostało zaprezentowane na Rys. 4.15.



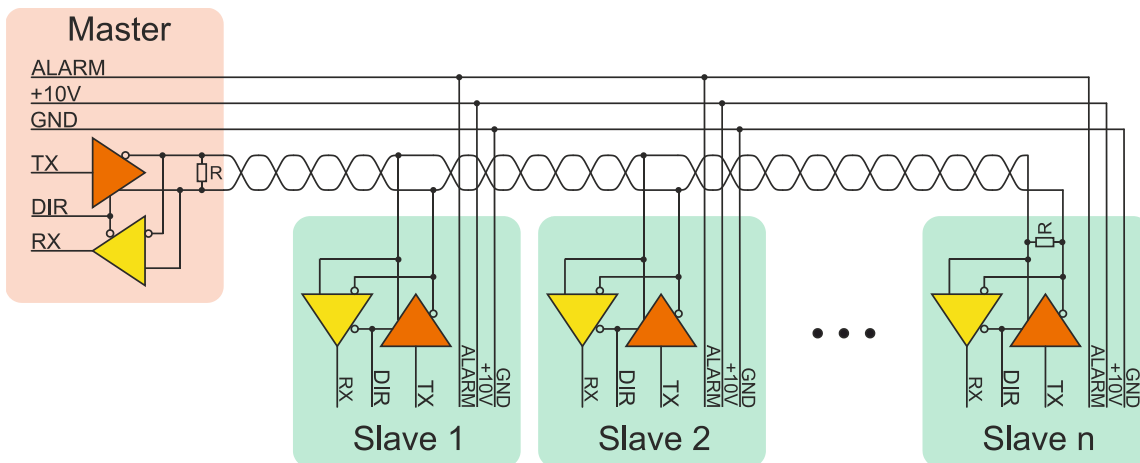
Rys. 4.15 Wizualizacja umieszczenia magnesu trwałego względem elementu pomiarowego AS5600

Podczas konstruowania ramienia robota starano się tak instalować elementy czujnika, by magnes umieszczony był w części ruchomej, a element pomiarowy w części nieruchomej. Takie rozwiązanie eliminuje problem degradacji przewodów czujnika, co jest istotną zaletą. Czujnik wyposażony jest również w detekcję braku magnesu lub jego zbyt dużego oddalenia, co zostało wykorzystane w systemie auto diagnostycznym robota. Taki mechanizm pozwala zasygnalizować awarię przed uruchomieniem robota lub w trakcie jego pracy.

Jednorazowo, podczas pierwszego uruchomienia robota, przeprowadzana jest kalibracja pozycji zerowej czujników bezwzględnych. W tym procesie ustawiane są wszystkie osie ramienia do pozycji zerowej, następnie wysyłany jest rozkaz do modułów Slave zapamiętania aktualnej pozycji jako zerowej. System sterowania zapamiętuje pozycję zerową niezależnie dla ustawienia na kończynę prawą i lewą. Fizycznie informacje o zapamiętanych pozycjach zerowych zapisane są w pamięci nieulotnej EEPROM modułu Slave.

4.10. Magistrala danych

Program sterujący pracą robota rehabilitacyjnego ARM-200 jest aplikacją uruchomioną w systemie Microsoft Windows 10, która działa na komputerze typu PC. Komunikacja z robotem zrealizowana została za pomocą połączenia kablowego Ethernet 100 Mbps. Komunikacja aplikacji sterującej, uruchomionej na komputerze PC, z modułami Slave realizowana jest za pomocą przesyłanych cyklicznie pakietów UDP do modułu Master, który później przesyła odebrane dane do modułów Slave. Każdy pakiet wysyłany z aplikacji komputerowej zawiera informację sterującą w postaci paczki danych dla wszystkich węzłów systemu. Następnie, moduł Master dzieli odebraną paczkę danych na mniejsze pakiety skierowane do odpowiednich węzłów Slave systemu. Po wysłaniu każdego pakietu do kolejnego węzła Slave, moduł Master czeka na odpowiedź modułu Slave. Fizyczna komunikacja pomiędzy modułem Master i modułami Slave zrealizowana została za pomocą magistrali danych RS-485 w postaci skręconej pary przewodów. Magistrali danych RS-485, przedstawionej na Rys. 4.16, towarzyszy linia alarmowa i zasilanie części pomiarowych modułów Slave. Takie rozwiązanie umożliwia komunikację z węzłami nawet po odłączeniu lub awarii zasilania napędów, co ułatwia przeprowadzenie diagnostyki robota w przypadkach awarii. W przypadku zgłoszenia awarii przez dowolny moduł Slave linią alarmową moduł Master ma możliwość odłączenia zasilania od modułów wykonawczych i tym samym zatrzymać pracę wszystkich napędów robota.



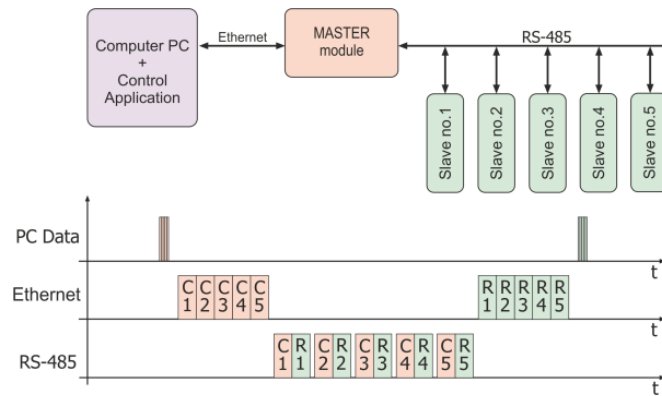
Rys. 4.16 Schematyczny opis magistrali robota, składającej się z magistrali danych RS-485, linii alarmowej i zasilania

Magistrala danych RS-485 jest magistralą typu half duplex, czyli w danej chwili możliwa jest komunikacja tylko w jedną stronę. Domyślnie magistrala RS-485 skonfigurowana jest we wszystkich modułach na nasłuch. Na Rys. 4.17 zostały zaprezentowane fizyczne sygnały zarejestrowane analizatorem stanów logicznych Saleae Logic Pro 8 z naniesionymi opisami. Pierwszy sygnał od góry „Master->Slave” reprezentuje dane nadawane przez moduł Master, następnie dane nadawane w odpowiedzi przez moduły Slave „Slave->Master” i sygnał „DIR” sterujący kierunkiem nadawania modułu Master. W zaprezentowanym przypadku moduł Slave nr 2 został odłączony, więc na linii sygnałowej „Slave->Master” w miejscu odpowiedzi modułu Slave nr 2 występuje luka, podczas której moduł Master czeka na odpowiedz modułu Slave.



Rys. 4.17 Sygnały logiczne magistrali RS-485 podczas wymiany danych między modulem Master i modułami Slave

Po zakończeniu sekwencji komunikacji z modułami Slave, moduł Master odsyła odebrane pakiety danych w postaci jednej paczki do aplikacji komputerowej, co zostało przedstawione na Rys. 4.18.



Rys. 4.18 Schemat komunikacji w systemie sterowania robotą

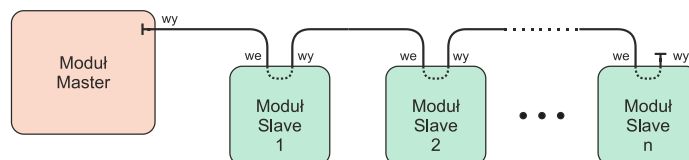
Komunikację pomiędzy modulem Master a komputerem PC, inicjuje komputer PC. Komunikacja odbywa się za pomocą grup pakietów wysyłanych jako blok danych zawierających ramki danych do wszystkich modułów Slave obecnych w systemie. Sterownik Master po odebraniu grupy ramek, rozsyła je sekwencyjnie do modułów Slave i jednocześnie pomiędzy kolejnymi transmisjami do modułów Slave oczekuje określony czas na uzyskanie odpowiedzi z danego modułu Slave. Natomiast moduł Master po odczekaniu czasu na uzyskanie odpowiedzi z ostatniego modułu Slave odsyła zgrupowane pakiety danych odpowiedzi modułów Slave do komputera PC. Budowę systemu komunikacji i sposobu wymiany danych pomiędzy elementami systemu sterowania przedstawia Rys.1.

Opóźnienia czasowe pomiędzy wysłaniem zapytań z aplikacji uruchomionej na komputerze PC a otrzymanej zwrotnie odpowiedzi składają się z dwóch elementów:

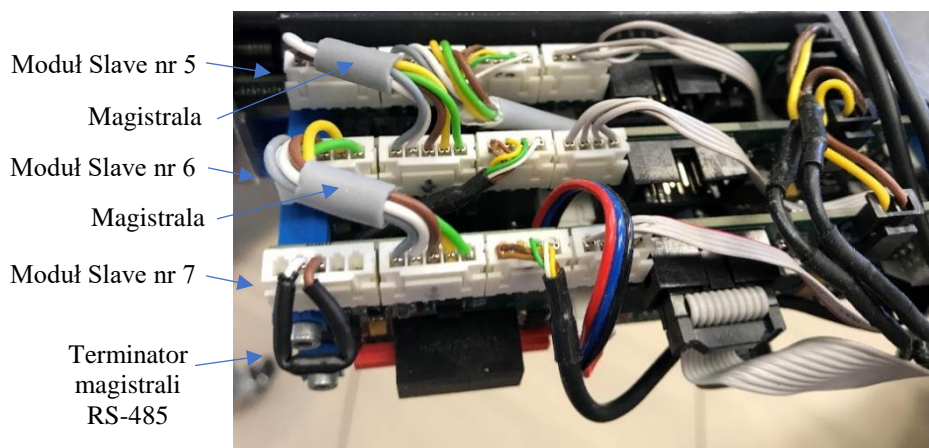
1. Zmiennego opóźnienia w komunikacji na magistrali Ethernet łączącej komputer PC z modulem Master. Na to opóźnienie będzie miał wpływ: typ zastosowanej magistrali (Ethernet) i obciążenie systemu operacyjnego PC w stosunku do możliwości sprzętowych komputera PC.
2. Stałego opóźnienia na wymianę informacji modułu Master z modułami Slave. Na czas tego opóźnienia ma wpływ wybór prędkości komunikacji magistrali szeregowej RS-485 (460800bps), ilości przekazywanych danych i liczby węzłów systemu.

4.11. Automatyczne adresowanie modułów Slave

Ważnym mechanizmem opracowanym na potrzeby systemu sterowania robota rehabilitacyjnego, jest automatyczne adresowanie modułów Slave. Jest to istotne podczas produkcji i serwisowaniu robota. Dla poprawnej i bezpiecznej pracy, każdy z węzłów Slave robota powinien mieć swój unikalny adres. Można to rozwiązać przez zapamiętanie w oprogramowaniu modułu Slave stałego adresu, ale rodzi to problem kompilowania oddzielnych wersji oprogramowania dla każdego adresu. Dodatkowy problem pojawia się podczas wymiany uszkodzonego modułu Slave. Wówczas przed instalacją nowego modułu, należy go zaprogramować odpowiednią wersją programu. Dla ułatwienia późniejszej produkcji i serwisu, problem adresowania został rozwiązany przez algorytm, który adresuje kolejne moduły Slave na podstawie kolejności podłączenia modułów Slave, licząc od modułu Master. Do automatycznego nadawania adresów wykorzystana została linia alarmowa magistrali a także fakt, że każdy z modułów posiada wejście i wyjście magistrali, co zostało przedstawione na Rys. 4.19. Dodatkowo, zastosowanie pomiędzy modułami Slave podziału magistrali na odcinki ułatwia serwis robota, ponieważ w przypadku uszkodzenia magistrali, wymianie podlega tylko uszkodzony odcinek.

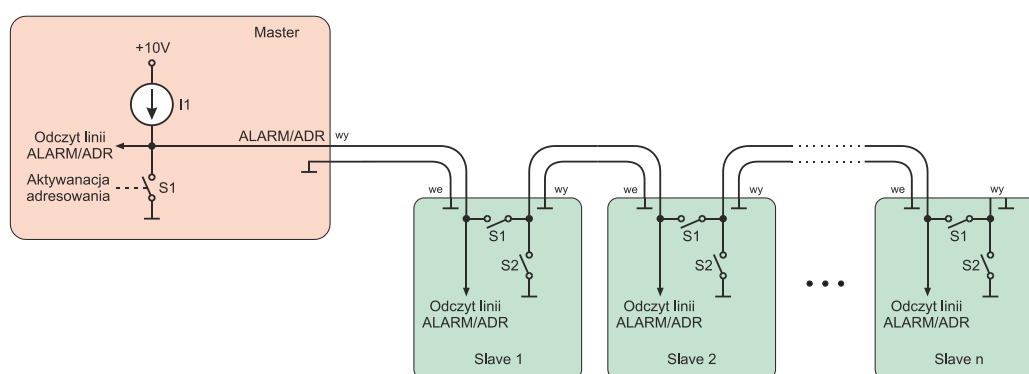


Rys. 4.19 Uproszczony schemat połączenia modułu Master z modułami Slave



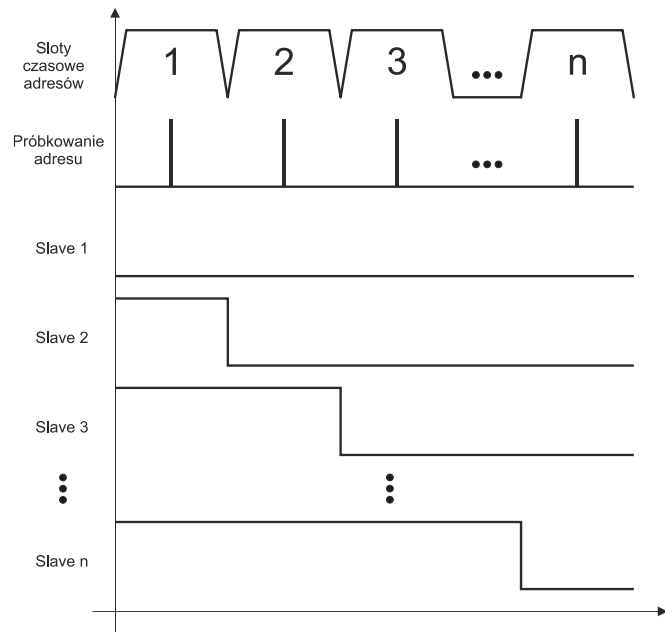
Rys. 4.20 Zdjęcie modułów Slave robota ARM-200 połączonych magistralą danych

Uproszczony schemat linii alarmowo-adresowej został przedstawiony na Rys. 4.21. Przed włączeniem zasilania modułów Slave, moduł Master zwiera klucz S1 do masy. Po upływie czasu połowy slotu czasowego przedstawionego na Rys. 4.22, moduł Slave nr 1 odczytuje stan niski linii adresowej i ustawia adres nr 1. Następnie, Slave nr 1 zwiera klucz S1, to powoduje przeniesienie stanu niskiego na wejście modułu Slave nr 2, który odczytuje stan niski w drugim slotie czasowym i przyjmuje adres nr 2 itd., w ten sposób stan niski przekazywany jest sekwencyjnie do kolejnych modułów Slave.



Rys. 4.21 Schemat działania linii alarmowo-adresowej

Podczas normalnej pracy klucz S1 w module Master i klucze S2 w modułach Slave pozostają rozwarte, natomiast klucze S1 w modułach Slave są zwarte. Linia alarmowo-adresowa łącząca moduł Master z modułami Slave jest podłączana źródłem prądowym w module Master. Podczas normalnej pracy zwarcie styku S2 w module Slave pozwala zasygnalizować innym węzłom i modułowi Master stan alarmowy, linia ta działa niezależnie od magistrali danych RS-485. Po włączeniu zasilania stan linii alarmowej nie jest przekazywany pomiędzy wejściem i wyjściem magistrali modułów Slave, co pozwala na realizację funkcji automatycznego adresowania. W pierwszym kroku, moduł Master włącza zasilanie wszystkich modułów Slave. Od tego momentu moduły Slave uruchamiają zliczanie czasu. Jak zostało przedstawione na Rys. 4.22, pierwszy moduł Slave odczytuje stan niski na linii ALARM, czyli stan niski pojawia się w pierwszym slotie czasowym, więc moduł Slave ustawia adres 1, następnie po zadany czasie 500ms moduł Slave 1 łączy wejście z wyjściem magistrali co powoduje przekazanie stanu niskiego na kolejny moduł Slave, który przyjmuje adres 2 itd.

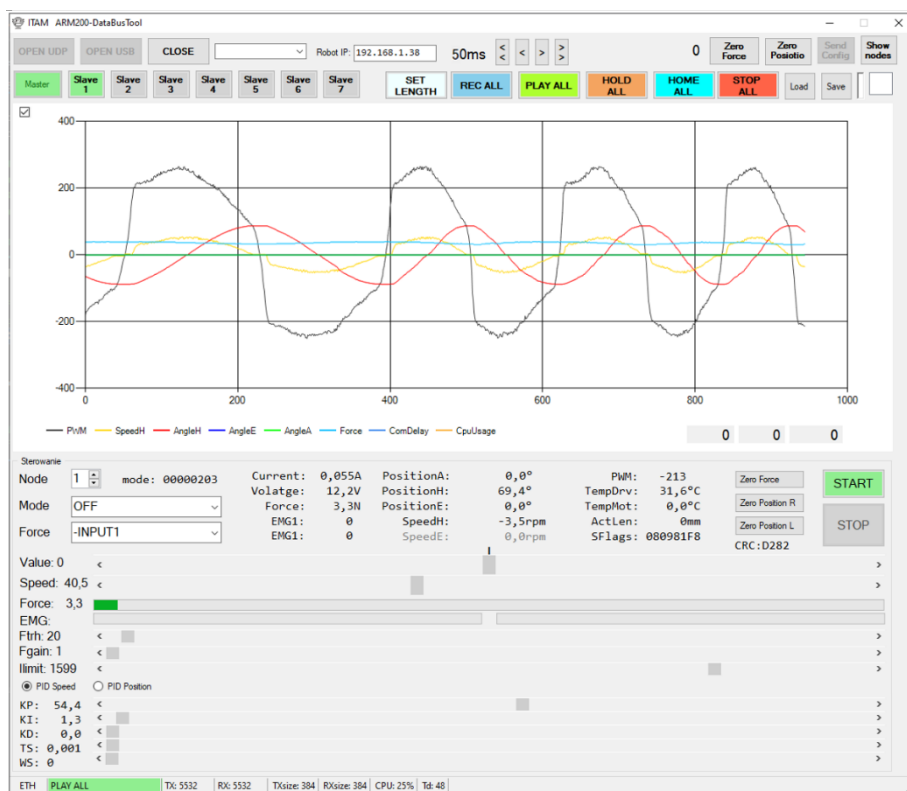


Rys. 4.22 Sygnały automatycznego adresowania modułów Slave

By zabezpieczyć się przed przypadkową zmianą adresu, w przypadku uszkodzenia innego modułu Slave, został wprowadzony mechanizm porównujący poprzednio uzyskany adres z obecnym. Uzyskany adres porównywany jest z adresem zapisanym uprzednio w nieulotnej pamięci EEPROM, jeżeli adres jest ten sam, to system przechodzi do dalszej pracy. Natomiast, jeżeli adresy się różnią to ustawiana i przesyłana do systemu jest flaga błędu o zmianie adresu, co zatrzymuje pracę robota.

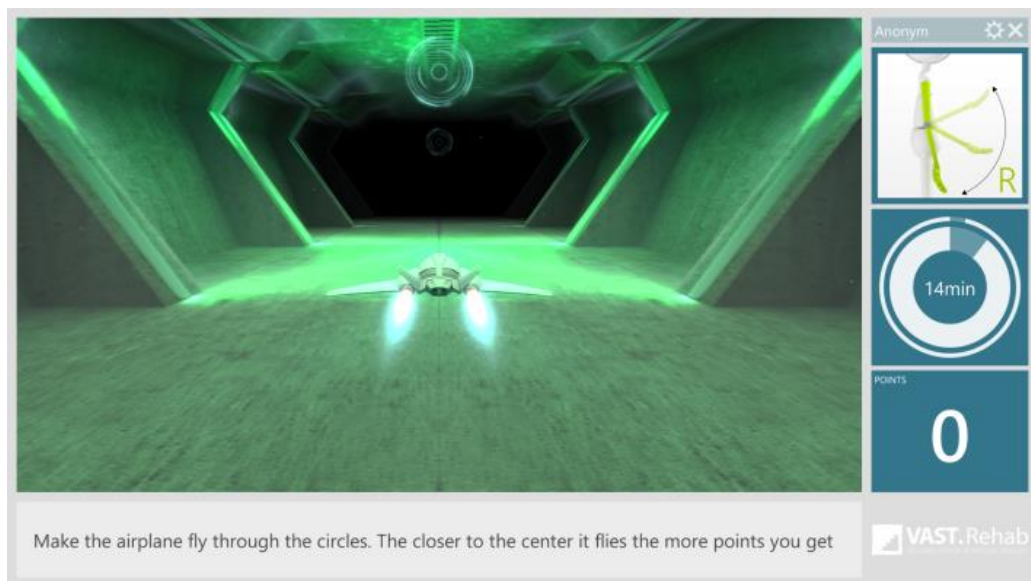
4.12. Aplikacja sterująca robotem

Ważnym elementem robota rehabilitacyjnego jest aplikacja komputerowa umożliwiająca kontrolę procesu rehabilitacji przez wybór ćwiczeń, rejestrację wyników rehabilitacji i kontrolę stanu robota. Podczas prac nad systemem sterowania robota na pierwszym etapie prac powstała tzw. aplikacja inżynierska „ARM-200 DataBusTool”, umożliwiająca w łatwy sposób przeprowadzenie prac rozwojowych i testów. Aplikacja bezpośrednio komunikuje się z modułem Master systemu sterowania robota i za pomocą tej aplikacji będą przeprowadzane eksperymenty, które zostały opisane w dalszej części niniejszej pracy. Ekran inżynierskiej aplikacji sterującej przedstawia Rys. 4.23.



Rys. 4.23 Ekran inżynierskiej aplikacji sterującej robotem „ARM-200 DataBusTool”.

Dodatkowo, w trakcie prac nad systemem sterowania robota rehabilitacyjnego została opracowana inna aplikacja inżynierska, która współpracuje z opracowaną dla robota ARM-200 biblioteką pośredniczącą pomiędzy robotem a aplikacją sterującą. Drugie rozwiązanie miało ułatwić integrację robota z oprogramowaniem sterującym firmy Brontes Processing, w którym robot stanowi interfejs użytkownika w różnych grach rehabilitacyjnych. Trening w tych grach polega sterowaniu np. wirtualnym statkiem kosmicznym lub samochodem. Podczas treningu oprogramowanie mierzy czas i skuteczność w realizacji zadanych celów. Oprogramowanie VAST.Rehab firmy Brontes Processing również archiwizuje wyniki w bazie danych pacjentów. Okno programu VAST.Rehab, zostało zaprezentowane na Rys. 4.24, w tym przykładowym scenariuszu rehabilitowana osoba steruje wysokością lotu statku kosmicznego za pomocą zginania i prostowania stawu łokciowego. Zadaniem gracza jest trafiać statkiem w okrągłe cele, które wyświetlane są na różnych wysokościach. Tego typu rehabilitacja staje się formą rozrywki, co jest szczególnie istotne z uwagi na żmudny i wielomiesięczny proces rehabilitacji. W wielu publikacjach potwierdzono istotny wpływ zastosowania gier na motywację pacjenta do wykonywania ćwiczeń rehabilitacyjnych kończyn górnych [13].

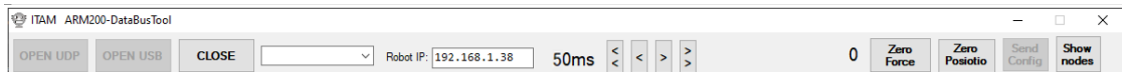


Rys. 4.24 Okno programu VAST.Rehab współpracujące z robotem ARM-200

Aplikacja „ARM-200 DataBusTool” powstała w środowisku Microsoft Visual Studio w języku C#, w systemie operacyjnym Microsoft Windows 10 i jest to aplikacja typu Windows Forms. Opracowana aplikacja inżynierska, umożliwia dowolne manipulowanie wybranymi parametrami np. regulatorów PID odpowiedzialnych za stabilizację prędkości czy pozycji elementów wykonawczych, pozwala na sterowanie całym robotem lub tylko wybranym węzłem wykonawczym robota.

W górnej linii (Rys. 4.25) inżynierskiej aplikacji „ARM-200 DataBusTool” od lewej umieszczone są trzy przyciski umożliwiające: połączenie z robotem przez Ethernet lub przez interfejs USB (interfejs USB nie jest obecnie oprogramowany), klawisz CLOSE służący do zamykania połączenia z robotem, niezależnie od interfejsu. Po prawej stronie klawiszy jest rozwijane okno wyboru wirtualnego portu COM, następnie jest numer IP robota (jest to parametr wymagany by połączyć się z robotem przez sieć Ethernet), następnie jest wyświetlana informacja o ustawionym okresie wymiany informacji z robotem, zmiana tego okresu możliwa jest za pomocą klawiszy strzałek. Podwojone strzałki pozwalają zmienić okres o 10 ms a pojedyncze o 1ms. Kolejnym parametrem wyświetlanym w tym fragmencie okna jest liczba próbek rejestrowanej trajektorii ruchu lub numer aktualnie odtwarzanej próbki przez liczbę wszystkich próbek treningu. Następnie widoczne są przyciski do zerowania toru pomiarowego siły i toru pomiarowego pozycji kątowej dla wszystkich węzłów jednocześnie. W dalszej

kolejności umieszczony jest przycisk uruchamiający przesłanie pakietów konfiguracyjnych i przycisk wyświetlania okna ze zbiorczymi parametrami np. sił dla wszystkich węzłów.



Rys. 4.25 Fragment okna aplikacji odpowiedzialny za nawiązanie połączenia z robotem

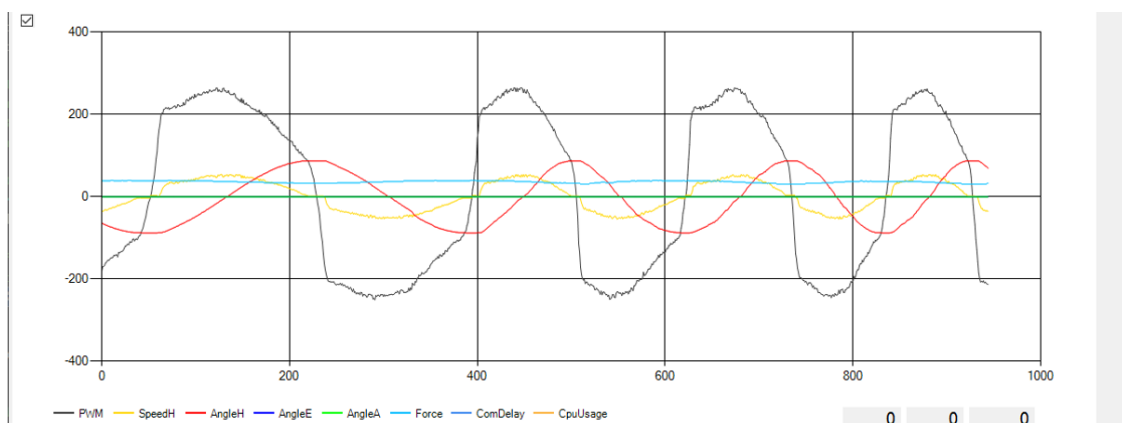
W linii poniżej, na Rys. 4.26, wyświetlane są symbole poszczególnych węzłów, a kolor zielony informuje o aktywnym połączeniu z danym węzłem. Następnie znajduje się grupa przycisków sterowania całym robotem. Przycisk „SET LENGTH” przełącza wszystkie węzły w tryb z aktywną opcją sterowania siłownikiem zmiany długości, w którym istnieje możliwość wysterowania siłowników za pomocą umieszczonych na obudowie fizycznych przycisków zmiany długości „+”, „-”. Przycisk „REC ALL” umożliwi uruchomienie trybu śledzenia we wszystkich węzłach robota. Natomiast przycisk „PLAY ALL” uruchamia zarejestrowany lub wgrany z dysku trening, dla wszystkich węzłów. Przycisk „HOLD ALL” zatrzymuje pracę wszystkich węzłów bez włączania hamulców, a przycisk „HOME ALL” uruchamia przejście ruch wszystkich napędów do pozycji „domowej” (ramię wzdłuż tułowia, przedramię zgięte w płaszczyźnie strzałkowej w stawie łokciowym pod kątem 90°). Przycisk „STOP ALL” zatrzymuje wszystkie napędy i uruchamia opcjonalne hamulce w węzłach.



Rys. 4.26 Kontrolki połączenia z węzłami robota i przyciski globalnego sterowania

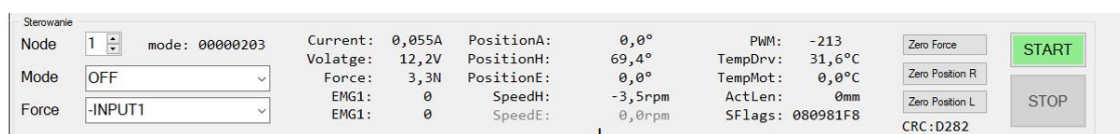
Przyciski „Load” i „Save” służą do odczytu lub zapisu treningu na dysk komputera.

Kolejny fragment okna aplikacji (Rys. 4.27) odpowiada za rysowanie wybranych parametrów, jak np. pozycja kątowa, PWM silnika, siła działająca na czujnik, zmierzony okres pomiędzy przesłanymi pakietami czy obciążenie procesora komputera.



Rys. 4.27 Fragment okna odpowiedzialny za rysowanie wykresów wybranych parametrów

Na Rys. 4.28 został przedstawiony fragment okna, w którym w polu „Node” można wybrać węzeł, którego parametry możemy wyświetlić lub którym chcemy sterować. Poniżej w polu „Mode” możemy wybrać tryb pracy węzła, a w polu „Force” możemy wybrać, z którego kanału pobieramy informacje o sile nacisku z możliwością negacji wyniku. Natomiast w środkowej części możemy obserwować chwilowe wartości istotnych parametrów pracy węzła. Po prawej stronie fragmentu okna umieszczono przyciski zerowania wartości siły i przyciski zerowania pozycji „domowej”: niezależnie dla prawej i lewej strony. Ostatnimi elementami jest przycisk „START” uruchamiający, wcześniej ustawiony w oknie Mode, tryb pracy i przycisk „STOP” zatrzymujący pracę wybranego węzła.



Rys. 4.28 Fragment okna odpowiedzialny głównie za wyświetlanie wartości liczbowych wybranych parametrów

Na Rys. 4.29 zaprezentowany jest fragment okna odpowiadający za ustawianie podstawowych parametrów pracy węzła. Jest to „Value”, który jest parametrem przełączanym w zależności od wybranego trybu pracy, np. w trybie stabilizacji pozycji jest wartością kątową pozycji, którą można zadać suwakiem, w trybie PWM jest to wartość wypełnienia sterująca silnikiem. Wartość „Speed” jest prędkością wykonywania ruchu. „Force” jest to jedynie wskaźnik graficznie informujący o przyłożonej do czujnika sile. Również „EMG” jest tylko dwukanałowym wskaźnikiem zmierzonego sygnału. Natomiast, „Ftrh” jest parametrem określającym próg siły, po przekroczeniu którego uruchamiany jest napęd i możliwe jest śledzenie ruchu w trybie śledzenia. „Fgain” jest parametrem umożliwiającym cyfrowe wzmocnienie reakcji na przyłożoną siłę w trybie śledzenia. Ostatnim parametrem jest „Limit”, który określa maksymalne wypełnienie sygnału PWM sterujące silnikiem.



Rys. 4.29 Fragment okna z suwakami umożliwiającymi zmianę podstawowych parametrów pracy węzła

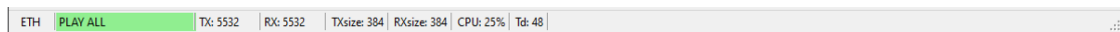
W części okna przedstawionego na Rys. 4.30, w zależności od ustawienia przycisków typu „radio button”: „PID Speed” lub „PID Position” możliwa jest modyfikacja parametrów regulatorów stabilizujących prędkość lub pozycję. Do dyspozycji mamy ustawienie: KP –

części proporcjonalnej, KI – części całkującej, KD – części różniczkującej i TS – czasu próbkowania regulatora z rozdzielczością 1ms. Dodatkowo jest parametr WS, który zostanie omówiony w dalszej części pracy, przy eksperymentach z napędem wyposażonym w przekładnię harmoniczną.



Rys. 4.30 Fragment okna, w którym jest możliwość zmiany parametrów regulatorów PID

W belce dolnej okna, przedstawionej na Rys. 4.31, umieszczone są informacje statusowe o połączeniu z robotem, trybie pracy, licznikach pakietów wysłanych i odebranych, a także o rozmiarach przesyłanych pakietów, obciążeniu procesora komputera i o zmierzonym interwale czasu z jakim prowadzona jest komunikacja aplikacji z komputerem.

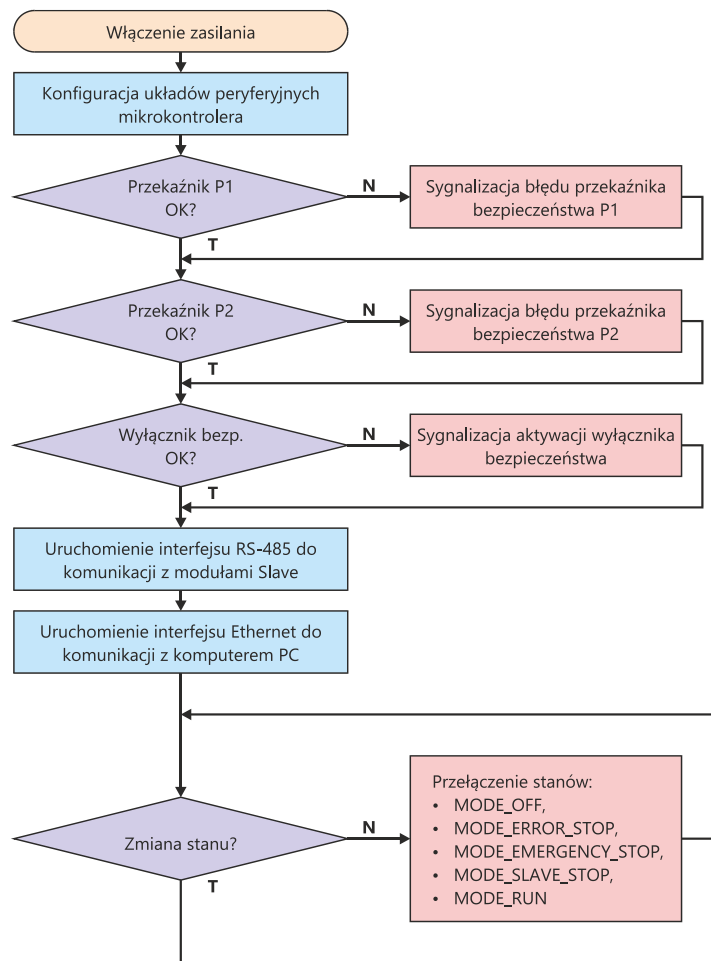


Rys. 4.31 Listwa dolna okna programu wyświetlająca informacje statusowe

4.13. Opis działania oprogramowania modułu Master

Po włączeniu zasilania, oprogramowanie modułu Master konfiguruje piny mikrokontrolera do obsługi elementów wykonawczych, sygnalizacyjnych i pomiarowych (Rys. 4.32). Poza funkcją komunikacyjną, moduł Master pełni bardzo ważną funkcję bezpieczeństwa w postaci sterowania przekaźnikami bezpieczeństwa. Rolą przekaźników bezpieczeństwa jest załączać lub odłączać zasilanie napędów robota w zależności od aktualnego stanu urządzenia (

Tab. 4.1).



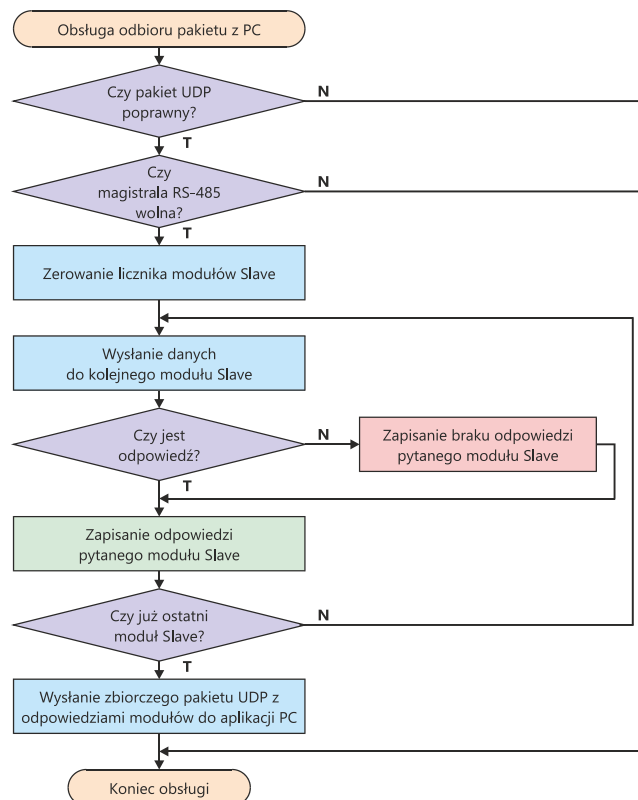
Rys. 4.32 Uproszczony schemat działania programu głównego modułu Master

Zastosowane przekaźniki bezpieczeństwa zostały wyposażone w wymuszone prowadzenie styków zgodnie z normą PN-EN 61810-1. Pozwala to wykryć stan tzw. sklejenia styków przekaźnika, co można odczytać sprawdzając stan styku pomocniczego przed zasilaniem cewki przekaźnika i właśnie taką sytuację wykrywa oprogramowanie wbudowane modułu Master. Zastosowanie zdublowanych szeregowo przekaźników bezpieczeństwa pozwala zapewnić skuteczne odłączenie zasilania w przypadku uszkodzenia jednego z przekaźników. Odłączenie zasilania jest możliwe drogą programową lub za pomocą wyłączników bezpieczeństwa, które bez pośrednictwa oprogramowania odłączają zasilanie cewek przekaźników bezpieczeństwa z możliwością odczytania stanu wyłączników bezpieczeństwa przez oprogramowanie modułu Master.

Tab. 4.1 Tabela możliwych stanów modułu Master

Lp.	Nazwa stanu	Opis
1.	MODE_OFF,	Wyłączenie zasilania napędów (tryb podstawowy)
2.	MODE_ERROR_STOP,	Wyłączenie napędów w wyniku wystąpienia awarii, np. wykrycia uszkodzenia jedno z przekaźników bezpieczeństwa
3.	MODE_EMERGENCY_STOP,	Wyłączenie napędów w wyniku naciśnięcia jednego z wyłączników bezpieczeństwa
4.	MODE_SLAVE_STOP,	Wyłączenie napędów w wyniku aktywacji linii Alarm przez jeden z modułów Slave
5.	MODE_RUN	Zasilanie napędów (tryb podstawowy)

Drugim podstawowym zadaniem modułu Master jest pośredniczenie w wymianie danych pomiędzy interfejsem Ethernet 100Mb i magistralą RS-485.

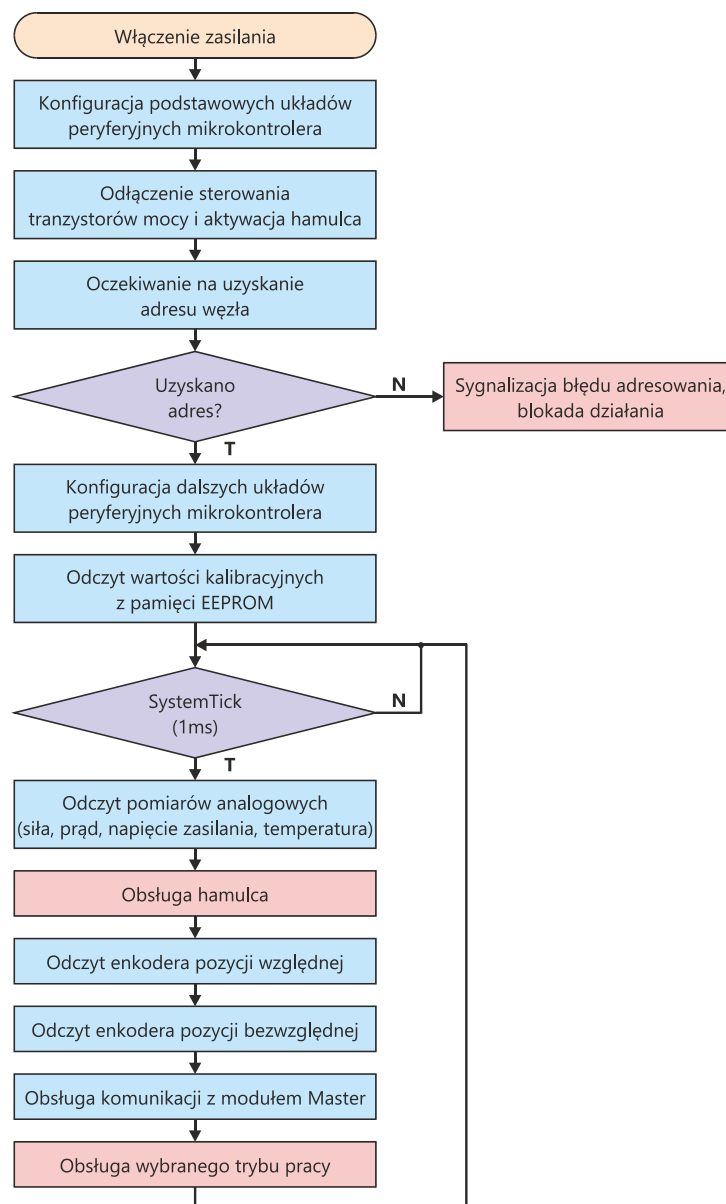


Rys. 4.33 Uproszczony schemat działania programu odbioru danych modułu Master

Transfer danych pomiędzy interfejsami zrealizowany został wykorzystując kanały DMA (ang.: Direct Memory Access) i przerwania, co pozwoliło uzyskać minimalne opóźnienia i ograniczenie obciążenia mikrokontrolera modułu Master.

4.14. Opis działania oprogramowania modułu Slave

Oprogramowanie wbudowane modułu Slave, przedstawione schematycznie na Rys. 4.34, po włączeniu zasilania uruchamia inicjalizację układów wewnętrznych i konfiguruje podstawowe peryferia modułu w stan bezpieczny, odłączając napięcia od faz silnika BLDC i aktywując opcjonalny hamulec napędu. Następnie, węzeł systemu przechodzi w tryb oczekiwania na zaadresowanie. Jeżeli w wymaganym czasie nie zostanie uzyskany adres, moduł Slave przechodzi do sygnalizacji i blokowania dalszego działania.



Rys. 4.34 Uproszczony schemat działania programu głównego modułu Slave

Jeżeli w wymaganym czasie zostanie uzyskany adres, moduł Slave przechodzi do konfigurowania dalszych układów peryferyjnych, takich jak: przetwornik analogowo-cyfrowy, interfejs szeregowy do obsługi komunikacji z modułem Master, interfejs I²C i interfejs szeregowy do komunikacji z opcjonalnym modułem sEMG. Następnie, korzystając z interfejsu I²C, odczytywana jest pamięć EEPROM, w której przechowywane są parametry konfiguracyjne, pozwalające np. odczytać w jakiej pozycji skonfigurowane jest ramię robota (lewa lub prawa strona) i offsety czujników pozycji bezwzględnej dla strony lewej i prawej. Kalibracja czujników pozycji bezwzględnej w każdym węźle przeprowadzana jest w fazie produkcji, dla każdej ze stron niezależnie, pozwala to ograniczyć błąd kalibracji pozycji wynikający z luzów na przekładniach mechanicznych. W następnym kroku, za pomocą informacji z czujnika pozycji bezwzględnej i wartości kalibracyjnych odczytanych z pamięci EEPROM, aktualizowany jest licznik pozycji względnej, który podczas dalszej pracy jest inkrementowany i dekrementowany za pomocą detekcji zbocz na sygnałach z czujników Halla wbudowanych w silnik BLDC. Warto tu zauważyć, że podstawowym zadaniem czujników Halla jest synchronizacja przełączania faz silnika BLDC podczas komutacji. W prezentowanym rozwiązaniu, mikrokontroler modułu Slave obsługuje programowo przełączanie faz silnika BLDC za pomocą przerwań od zmian stanów na wejściach trzech czujników Halla.

Po pełnym skonfigurowaniu mikrokontrolera modułu Slave, program główny oczekuje na flagę zapaloną co 1ms, która synchronizuje zadania obsługiwane cyklicznie podczas pracy modułu. Podczas każdego cyklu taktowane są pomiary analogowe, odczytywane są czujniki pozycji kątowej, analizowane są dane odbierane z modułu Master i realizowane są przesłane rozkazy i wybrane tryby pracy modułu. Niezależnie od realizacji zadań taktowanych cyklicznie timerem SysTick (1ms), moduł Slave realizuje komunikację z modułem Master za pomocą kanałów DMA i przerwań.

Jedną z istotnych cech modułu Slave jest możliwość przełączania trybów pracy. Obecnie dostępna jest dość długa lista możliwych trybów, zaprezentowana w Tab. 4.2, z których część służy jedynie do celów testowych. Są też tryby pracy, które są uruchamiane jednorazowo w charakterze rozkazów, jak np. tryby: **MODE_CAL_FORCE**, **MODE_CAL_POSITION_REL**, **MODE_CAL_POSITION_ABS_R**, **MODE_CAL_POSITION_ABS_L**, które służą do wydawania jednorazowych zleceń zerowania siły czy pozycji. Natomiast tryby: **MODE_SET_SIDE_LEFT** i **MODE_SET_SIDE_RIGHT** służą do sygnalizacji zakończenia sekwencji ruchów odpowiedzialnych za przełączenie robota ze strony prawej na lewą i odwrotnie.

Tab. 4.2 Tabela możliwych trybów pracy modułu Slave

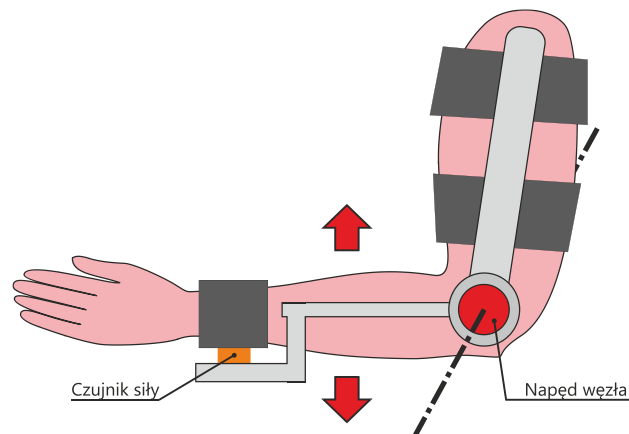
Lp.	Nazwa trybu pracy	Opis
6.	MODE_OFF	Wyłączenie napędu (tryb podstawowy)
7.	MODE_OFF_LENGTH_MAN	Wyłączenie napędu, możliwość ręcznego sterowania (za pomocą klawiszy) siłownikiem zmiany długości.
8.	MODE_OFF_LENGTH_SET	Wyłączenie napędu, sterowanie zdalne siłownikiem zmiany długości
9.	MODE_POSITION	Sterowanie napędem przez zadanie pozycji (tryb podstawowy - trzymanie pozycji / odtwarzanie ruchu / ćwiczenia pasywne)
10.	MODE_FORCE_POSITION	Sterowanie zmianą pozycji za pomocą czujnika siły, który śledzi ruch kończyny (tryb podstawowy - uczenie trajektorii / ćwiczenia aktywne)
11.	MODE_CAL_FORCE	Zerowanie wejść pomiarowych siły
12.	MODE_CAL_POSITION_REL	Zerowanie pozycji kątowej względnej (przepisanie pozycji bezwzględnej do enkodera względnego)
13.	MODE_CAL_POSITION_ABS_R	Zerowanie pozycji kątowej HOME bezwzględnej dla strony prawej i zapis pozycji do pamięci EEPROM
14.	MODE_CAL_POSITION_ABS_L	Zerowanie pozycji kątowej HOME bezwzględnej dla strony lewej i zapis pozycji do pamięci EEPROM
15.	MODE_PWM	Sterowanie napędem przez zadanie PWM (tylko do testów)
16.	MODE_SPEED	Sterowanie napędem przez zadanie prędkości (tylko do testów)
17.	MODE_FORCE_PWM	Sterowanie napędem przez czujnik siły, który zadaje wypełnienie PWM silnika (tylko do testów)
18.	MODE_POSITION_NLIM	Zmiana pozycji bez ograniczenia zakresu ruchu, tryb tylko do zmiany strony LEWA \leftrightarrow PRAWA
19.	MODE_FORCE_POSITION_NLIM	Sterowanie pozycją za pomocą czujnika siły bez ograniczenia zakresu ruchu (tylko do ustawiania pozycji przed kalibracją czujników pozycji bezwzględnej)
20.	MODE_SET_SIDE_LEFT	Ustawienie pozycji na stronę lewą (wpis do EEPROM)
21.	MODE_SET_SIDE_RIGHT	Ustawienie pozycji na stronę prawą (wpis do EEPROM)

Podczas normalnej pracy, moduł Slave realizuje dwa tryby pracy, jest to tryb śledzenia ruchu (**MODE_FORCE_POSITION**) i tryb odtwarzania ruchu (**MODE_POSITION**).

Tryb śledzenia ruchu

Tryb śledzenia ruchu umożliwia intuicyjne programowanie trajektorii ruchowych a także umożliwia realizację treningów aktywnych, w których robot śledzi ruchy kończyny za pomocą czujników siły, dążąc do uzyskania siły wywieranej na czujniki, w idealnym przypadku, równej zero. W przypadku uproszczenia robota do jednego stopnia swobody, przedstawionego na Rys. 4.35, próba zgięcia kończyny w stawie łokciowym będzie powodować wywieranie siły na

czujnik siły umieszczony pod uchwytem przedramienia. Algorytm śledzenia ruchu zmierzy wywieraną siłę i uruchomi napęd umieszczony w osi stawu łokciowego w takim kierunku by wywieraną siłę zmniejszyć. Z punktu widzenia realizacji treningów aktywnych, ważną cechą algorytmu sterującego jest możliwość zadawania progu siły, po przekroczeniu którego robot uruchamia napęd i zaczyna śledzić ruch kończyny. Takie rozwiązanie pozwala w łatwy sposób dostosowywać stopień obciążenia do stanu rehabilitowanego pacjenta.

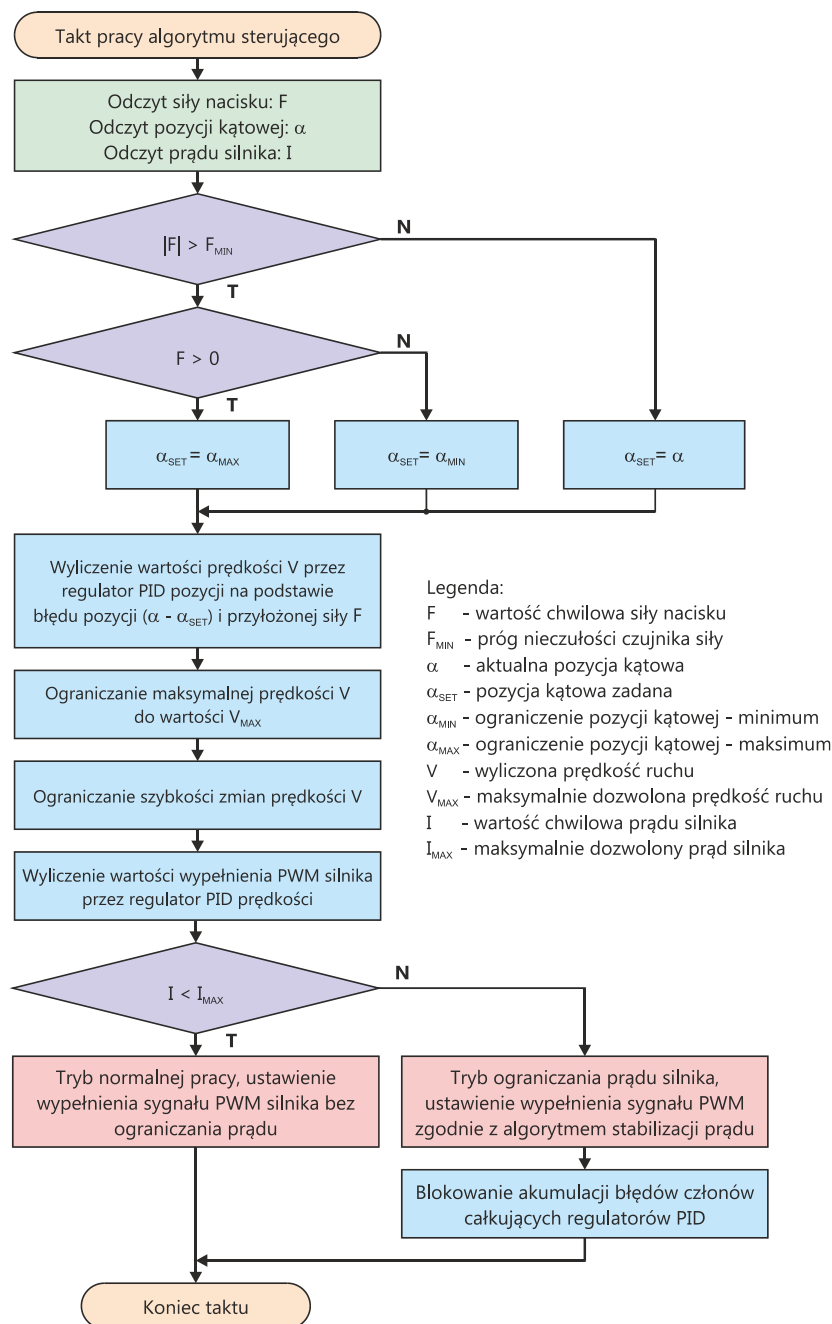


Rys. 4.35 Schemat działania pojedynczego węzła robota

W przypadku rejestracji ruchu, próg nieczułości jest ustawiony na minimalnym poziomie, zapewniającym dużą łatwość w wykonywaniu ruchu ramieniem. Wprowadzenie progu nieczułości układu śledzenia jest wymagane z uwagi na występujące w układzie pomiarowym szumy, zakłócenia i wywierane siły wynikające z przyjętych rozwiązań mechanicznych. Przykładowym źródłem błędów siły, może być ciężar uchwytu kończyny zamocowanego do czujnika siły po stronie pacjenta. Innym źródłem powstawania błędów pomiaru siły, mogą być naprężenia mechaniczne w miejscach, gdzie uchwyt kończyny podparty jest przez dwa czujniki siły. Wówczas nawet minimalne odkształcenie podstawy czujników, może być dodatkowym źródłem błędów w torze pomiarowym siły.

Poza czujnikiem siły, istotnymi elementami pomiarowymi są czujniki pozycji kątowej wężła. Podczas procesu uruchamiania systemu sterowania, z aplikacji sterującej uruchomionej w komputerze sterującym robotem, przesyłane są rozkazy konfiguracyjne każdy z węzłów. Jednym z bardzo istotnych parametrów są zakresy kątowe pracy danego wężła, które ograniczają zakres ruchu danej osi swobody. Na Rys. 4.36, opisującym tryb śledzenia ruchu (MODE_FORCE_POSITION) są to kąty: α_{MIN} i α_{MAX} . W tym trybie pracy, jeżeli zmierzona siła nie przekracza wartości progowej F_{MIN} węzeł pozostaje w aktualnej pozycji bez ruchu. Natomiast

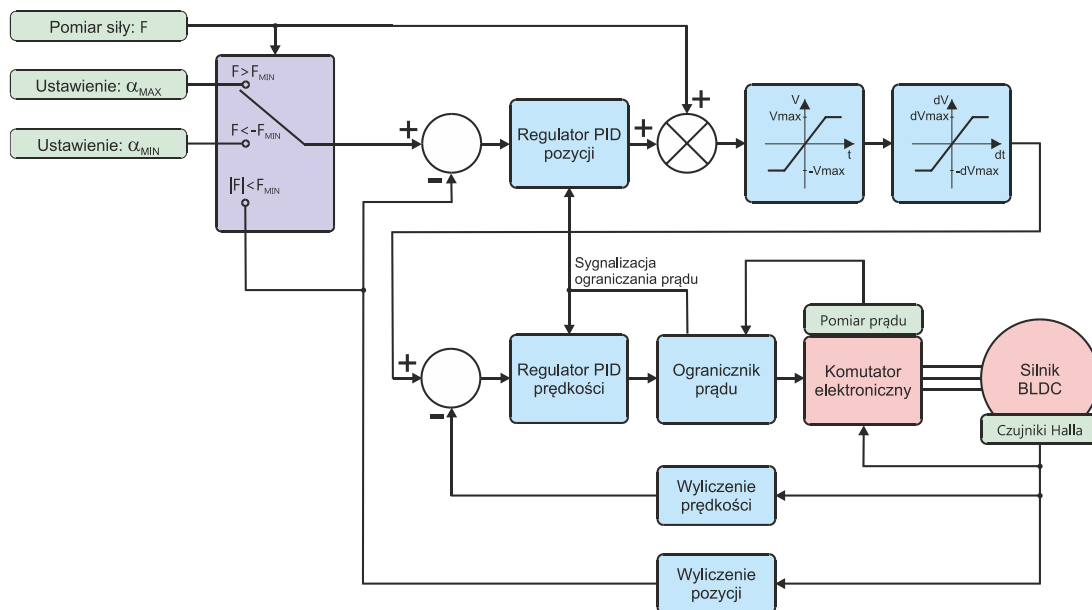
jeżeli wywarta siła jest większa od F_{MIN} węzeł uruchamia napęd by wykonać ruch do pozycji α_{MIN} lub α_{MAX} w zależności od znaku siły F . Prędkość ruchu jest wprost proporcjonalna do przyłożonej siły z ograniczeniem dopuszczalnej siły F_{MAX} i dopuszczalnego przyrostu prędkości. W tym trybie prędkość ruchu spada do zera w przypadku osiągnięcia granicy dopuszczalnego zakresu ruchu.



Rys. 4.36 Opis algorytmu sterowania w trybie śledzenia ruchu

Dodatkowym zabezpieczeniem modułu Slave w tym trybie jest ograniczenie prądu silnika, które jest powiązane z blokowaniem akumulowania błędów regulatorów PID. Jest to bardzo

ważny mechanizm ograniczający ryzyko gwałtownego ruchu ramienia robota po ustąpieniu przeszkody blokującej ruch. Pracę trybu śledzenia można również przedstawić w innej postaci schematu blokowego, w którym jest łatwiej prześledzić źródła i połączenia sygnałów biorących udział w wypracowywaniu sygnału sterującego napędem wykonawczym robota. Jak widać na Rys. 4.37, ruch w trybie śledzenia jest możliwy tylko w wypadku wywarcia nacisku na czujnik siły, siłą większego niż F_{MIN} pod warunkiem, że ramię robota nie znajduje się w skrajnych pozycjach dowolnego zakresu ruchu. W pozycjach skrajnych (α_{MIN} lub α_{MAX}), ruch jest możliwy tylko w kierunku przeciwnym do ograniczenia. Prędkość ruchu jest wprost proporcjonalna do przyłożonej siły ponad próg nieczułości siły. Jeżeli przyłożona siła wywoła przekroczenie prędkości o wartości V_{MAX} , to prędkość będzie limitowana do prędkości V_{MAX} . Dodatkowo, algorytm sterowania ogranicza prędkość narostu siły.

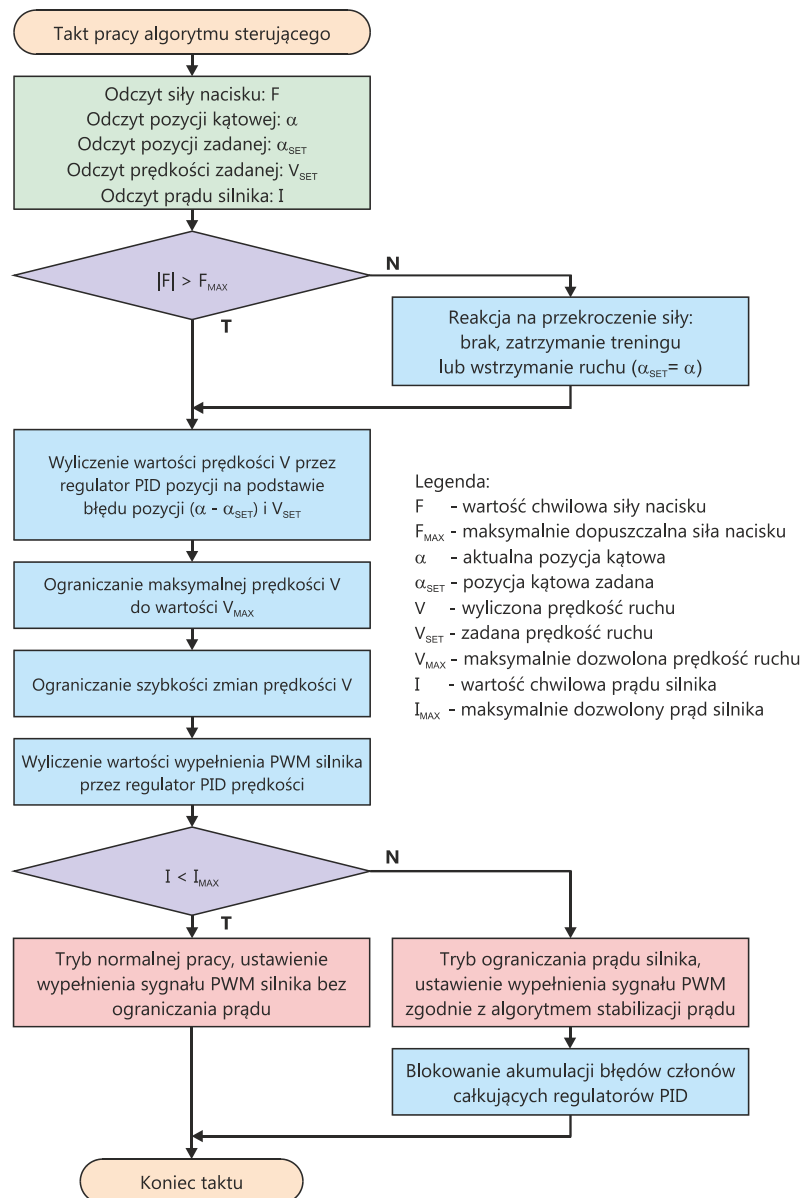


Rys. 4.37 Schemat blokowy sterowania w trybie śledzenia ruchu

Tak wypracowana wartość prędkości podawana jest na regulator PID stabilizujący prędkość ruchu, w wyniku czego regulator wypracowuje wartość wypełnienia sygnału PWM, który jest podawany do układu odpowiedzialnego za komutację faz silnika BLDC za pomocą tranzystorów mocy. Prąd zasilania komutatora jest mierzony i przekroczenie wartości dozwolonej powoduje przejście układu stabilizacji prędkości w tryb stabilizacji prądu. Dodatkowo, przekroczenie prądu blokuje akumulowanie błędów przez człon całkujący regulatora PID prędkości.

Tryb odtwarzania ruchu

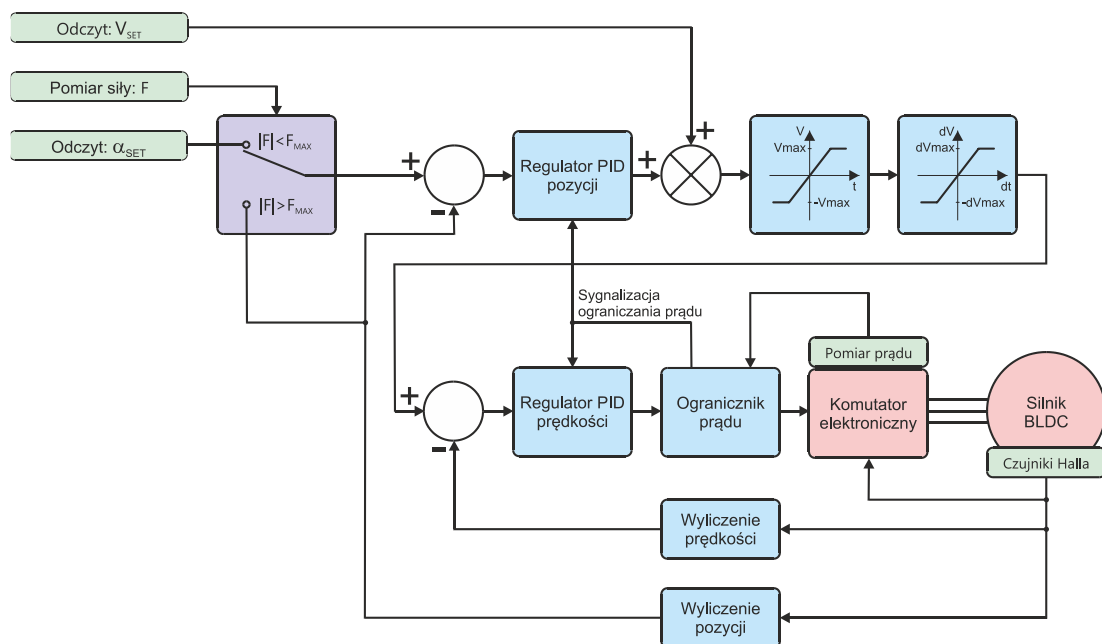
Drugim podstawowym trybem pracy węzła robota jest tryb odtwarzania ruchu, w którym robot wykonuje ruch do zadanej pozycji α_{SET} zadaną prędkością V_{SET} . Ten tryb, przedstawiony na Rys. 4.38, umożliwi realizację treningów biernych, w których ruch może być wygenerowany syntetycznie lub też odtworzony z poprzednio zarejestrowanego ruchu w trybie śledzenia.



Rys. 4.38 Opis algorytmu sterowania w trybie odtwarzania ruchu

W tym trybie pracy robota, bardzo ważnym zagadnieniem jest wybór reakcji na przekroczenie siły F_{MAX} . Podczas pracy z pacjentami z niedowładami kończyn może wystąpić skurcz mięśni, powodujący duży dyskomfort lub ból. W takim wypadku należy zapewnić zatrzymanie ruchu

robota. Drugą postacią opisu pracy modułu Slave w trybie odtwarzania ruchu jest schemat blokowy na Rys. 4.39. W tym trybie większość algorytmu pracy powielona jest z trybu śledzenia ruchu, zasadniczą różnicą jest sposób pozyskania parametrów wejściowych pozycji i prędkości. W trybie odtwarzania, te parametry są sekwencyjnie podawane na wejścia przez magistralę sterującą i jedynie przekroczenie wywieranej siły na czujnik nacisku lub przekroczenie dopuszczalnego prądu silnika może wstrzymać odtwarzanie sekwencji kolejnych par próbek pozycji i prędkości.

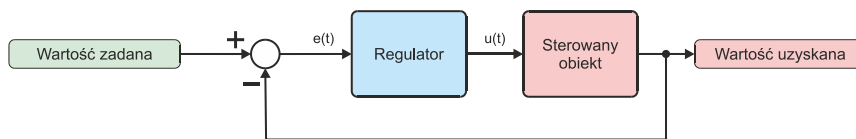


Rys. 4.39 Schemat blokowy sterowania w trybie odtwarzania ruchu

W trybie odtwarzania można zastosować dwie strategie zadawania ruchu. Pierwsza strategia polega na zadaniu docelowej pozycji kątowej i ustawienie prędkości z jaką ramię robota ma tą pozycję osiągnąć. Drugą strategią jest ustawienie stałej wysokiej prędkości ruchu i zadawanie sekwencyjne drobnych zmian pozycji.

Regulator prędkości PID

Bardzo ważnym elementem, opisanych wyżej trybów: śledzenia ruchu i odtwarzana ruchu, jest regulator prędkości, którego zadaniem jest stabilizacja określonego parametru, w tym wypadku prędkości ruchu ramieniem robota. W uproszczeniu, zadaniem regulatora jest wypracować taką wartość sterującą obiektem by na wyjściu obiektu uzyskać oczekiwaną wartość, zadaną na wejściu. Regulator wylicza wartość wyjściową $u(t)$ na podstawie sygnału uchybu $e(t)$, który jest różnicą pomiędzy wartością zadaną a wartością uzyskaną na wyjściu (Rys. 4.40).



Rys. 4.40 Uogólniony schemat pracy regulatora

W prezentowanej pracy, do stabilizacji prędkości wykorzystano regulator proporcjonalno-całkująco-różniczkujący. Skrócowa nazwa tego regulatora (PID) pochodzi do angielskich słów: Proportional, Integral, Derivative. Postać idealnego regulatora PID można zapisać wzorem [46, s. 70–81]:

$$u(t) = k_p \left(e(t) + \frac{1}{T_i} \int_0^t e(\tau) d\tau + T_d \frac{de(t)}{dt} \right) \quad (4.1)$$

gdzie:

- $u(t)$ – sygnał wyjściowy regulatora,
- $e(t)$ – sygnał wejściowy regulatora (sygnał uchybu),
- k_p – współczynnik wzmocnienia,
- T_i – czas całkowania,
- T_d – czas różniczkowania.

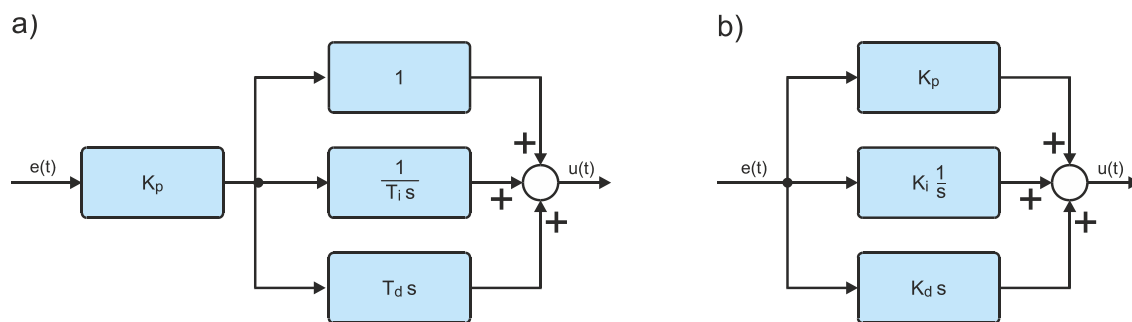
Natomiast transmitancja operatorową (Laplace'a) regulatora PID można zapisać w postaci przedstawionej poniżej [47, s. 77]:

$$C(s) = K_p \left(1 + \frac{1}{T_i s} + T_d s \right) \quad (4.2)$$

gdzie:

- $U(s)$ – transformata Laplace'a sygnału wyjściowego,
- $E(s)$ – transformata Laplace'a uchybu,
- K_p – współczynnik wzmocnienia,
- T_i – czas całkowania (czas zdwojenia),
- T_d – czas różniczkowania (czas wyprzedzenia).

Budowę regulatora PID można również opisać graficznie w postaci trzech członów. Klasyczny regulator PID ma budowę kaskadową jak na Rys. 4.41a, natomiast bardziej intuicyjną wersją jest struktura równoległa, która pozwala niezależnie modyfikować każdy z parametrów. Po wyznaczeniu parametrów regulatora za pomocą opisanej metody, możemy je niezależnie modyfikować ręcznie.



Rys. 4.41 Budowa regulatora PID: a - klasyczna kaskadowa, b- równoległa

W wersji równoległej postać operatorowa regulatora przyjmuje postać:

$$C(s) = K_p + K_i \frac{1}{s} + K_d s \quad (4.3)$$

gdzie:

- K_p – współczynnik wzmocnienia,
- K_i – współczynnik całkowania $= \frac{K_p}{T_i}$,
- K_d – współczynnik różniczkowania $= K_p T_d$.

Wzmocnienie członu proporcjonalnego regulatora jest ustawiane za pomocą parametru K_p . Jego powiększanie, podczas sterowania obiektem inercyjnym, umożliwiłoby zmniejszanie błędów w stanie ustalonym regulatora. Jednak wzrost wartości K_p prowadzi również do ryzyka utraty stabilności regulatora i pojawienia się oscylacji na jego wyjściu [46, s. 51]. By móc pozostawić niższą wartość wzmocnienia członu proporcjonalnego i jednocześnie zredukować błąd regulacji w stanie ustalonym wykorzystywany jest człon całkujący, w którym współczynnik K_i decyduje o szybkości zerowania błędów na wyjściu regulatora, akumulując błąd w czasie, co jest działaniem stosunkowo wolnym. Warto w tym miejscu również wspomnieć, że w praktycznych rozwiązaniach, na czas nasycenia wyjścia regulatora lub aktywacji ograniczania prądu silnika, należy wstrzymać całkowanie. Brak takiego mechanizmu prowadzi zazwyczaj do przeregulowania regulatora po ustąpieniu ograniczenia. Taki mechanizm w literaturze nosi angielską nazwę układu Anti-Windup [48]. Natomiast współczynnik K_d steruje reakcją członu

różniczkowego regulatora. Co sprowadza się do kontroli reakcji regulatora na nagłą zmianę sygnału wejściowego, a tym samym umożliwia reagować na błędy dynamiczne, chociaż w niektórych przypadkach, gdy zależy nam na łagodnej pracy regulatora, człon ten jest wyłączany przez wyzerowanie współczynnika K_d .

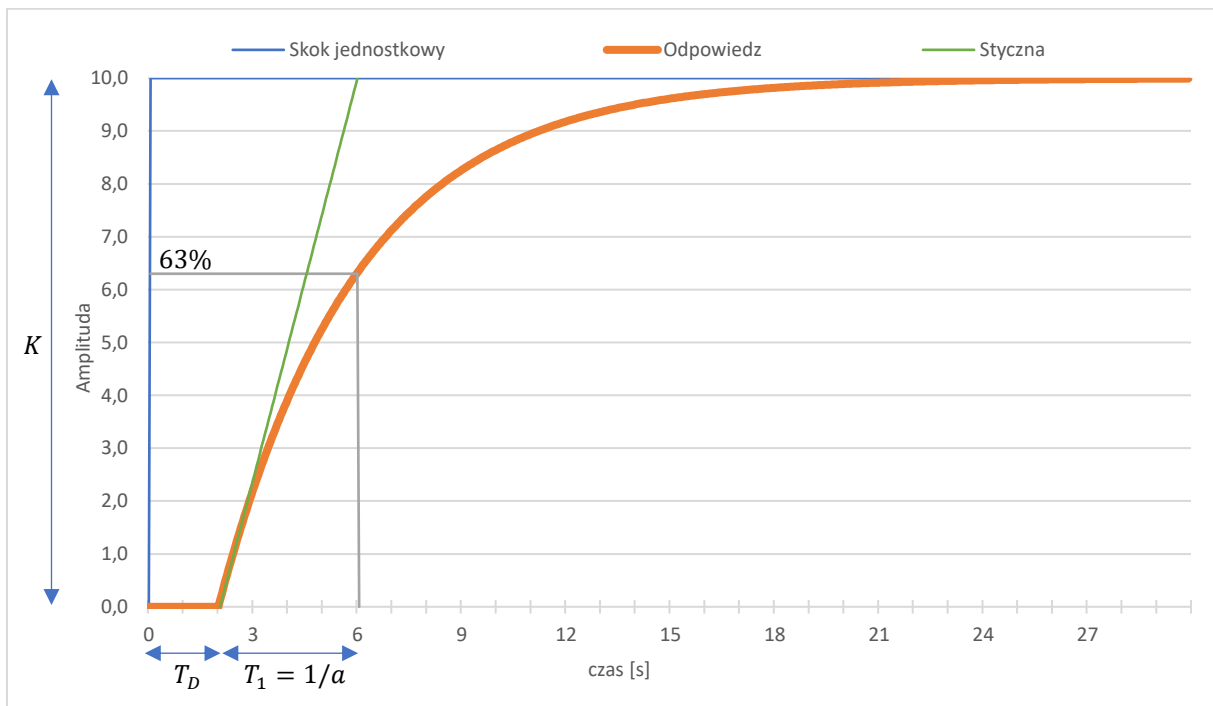
By dobrać odpowiednie nastawy regulatora PID należy zidentyfikować sterowany obiekt. W tym przypadku obiektem jest silnik prądu stałego z obciążeniem mechanicznym, w którym szybkość zmiany prądu jest znacznie większa niż szybkość zmian prędkości obrotowej, czyli można uznać, że prąd natychmiast dostosowuje się do zmian napięcia, a cała dynamika układu jest w uproszczeniu funkcją zmiany prędkości od zmiany napięcia. W takim przypadku, odpowiedź obiektu na skok jednostkowy może zostać opisywana równaniem różniczkowym pierwszego rzędu, zgodnie z równaniem poniżej [47, s. 79]:

$$G(s) = \frac{K e^{-sT_D}}{1 + sT_1} \quad (4.4)$$

gdzie:

- K – wzmacnienie układu (statyczne),
- T_D – czas opóźnienia obiektu,
- T_1 – stała czasowa obiektu.

Rys. 4.42 przedstawia przykładową odpowiedź obiektu pierwszego rzędu na skok jednostkowy, gdzie: $K = 10$, $T_D = 2$ i $T_1 = 4$.



Rys. 4.42 Odpowiedź obiektu na skok jednostkowy

By uzyskać stałą czasową obiektu należy wyznaczyć styczną do początkowej części wykresu odpowiedzi. Miejsce przecięcia stycznej z wartością ustaloną odpowiedzi wyznacza stałą czasową obiektu pierwszego rzędu T_1 , która opisywana jest również jako $1/a$ lub τ . Jest to czas, po którym amplituda odpowiedzi na skok jednostkowy osiąga wartość 63%. Po podstawieniu odczytanych z wykresu parametrów, odpowiedź obiektu można opisać równaniem:

$$G(s) = \frac{10e^{-s^2}}{1 + s4} \quad (4.5)$$

Znając już parametry obiektu, można przejść do doboru parametrów regulatora PID. Dobór nastaw regulatora jest bardzo ważnym i jednocześnie trudnym zagadnieniem, które może zostać przeprowadzone ręcznie lub automatycznie [46, s. 103]. W naszym przypadku parametry regulatora zostaną wyliczone ręcznie. Do najpopularniejszej metody doboru parametrów należy metoda Ziglera-Nicholsa, w której do identyfikacji obiektu można zastosować badanie z otwartą lub zamkniętą pętlą sprzężenia regulatora. W naszym przypadku przeprowadzono badanie odpowiedzi obiektu na skok jednostkowy z otwartą pętlą sprzężenia zwrotnego. Innym podejściem, przy zamkniętej pętli sprzężenia zwrotnego, jest nastawienie regulatora PID na działanie tylko części proporcjonalnej. Następnie, zwiększa się stopniowo współczynnik proporcjonalny regulatora do wystąpienia oscylacji. W ten sposób można wyznaczyć okres oscylacji i krytyczny współczynnik wzmocnienia, na podstawie których można wyznaczyć parametry regulatora PID [46, s. 110]. Natomiast, w Tab. 4.3 zostały podane wzory do wyliczenia współczynników wybranego regulatora (P, PI, PID) wg metody Ziglera-Nicholsa [47, s. 77] na podstawie odpowiedzi obiektu na skok jednostkowy.

Tab. 4.3 Wyliczenia współczynników regulatora wg metody Ziglera-Nicholsa

Typ regulatora	Współczynniki regulatora		
	K_p	T_i	T_d
P	$\frac{T_1}{KT_D}$		
PI	$\frac{0,9 T_1}{KT_D}$	$3,3 T_D$	
PID	$\frac{1,2 T_1}{KT_D}$	$2 T_D$	$0,5 T_D$

Istotną zaletą metody Ziglera-Nicholsa jest prostota implementacji. Niestety, tą metodą uzyskuje się dość duże przeregulowania (25%), które nie są pożądane w naszym przypadku. W literaturze można spotkać informacje, że bardzo małe przeregulowania może uzyskać w metodzie Lambda [47, s. 82], [49], która ma zastosowanie tylko do regulatora typu PI. W przypadku sterowania prędkością napędu takie ograniczenie nie stanowi wady, a wręcz może być uznane za zaletę, ponieważ nagłe przyspieszanie ramienia robota, generowane przez człon różniczkowy, mogłoby wywoływać dyskomfort dla rehabilitowanego pacjenta. W metodzie Lambda, współczynniki regulatora PI wyznaczone są na podstawie wzorów przedstawianych w Tab. 4.4, parametry T_1 i T_D można odczytać z Rys. 4.42.

Tab. 4.4 Wyliczenia współczynników regulatora wg metody Lambda

Typ regulatora	Współczynniki regulatora		
	K_p	T_i	T_d
PI	$\frac{T_1}{K(\lambda + T_D)}$	T_1	

Parametr λ , występujący we wzorze przedstawionym w tabeli powyżej, jest ustalany w zależności od oczekiwanej dynamiki regulatora. Dla szybkich procesów wartość parametru λ jest ustawiana mała, a dla wolniejszych większa. Zgodnie z literaturą [50], można przyjąć, że przeregulowania praktycznie nie występują dla wartości $\lambda \geq \left(\frac{1}{2} + \sqrt{2}\right) T_1$, co dla przypadku granicznego można oszacować na $\lambda = 1,9 T_1$. Należy również zwrócić uwagę na fakt, że rozpatrywany w tym miejscu przypadek jest wyidealizowany, w normalnej pracy robota rehabilitacyjnego, napęd będzie obciążony obciążeniem zmiennym. Będzie to wynikało z pozycji ramienia robota (o określonej masie) względem wektora grawitacji, a także ze zmiennego nacisku kończyny pacjenta na ramię robota. Dlatego, po wyznaczeniu parametrów regulatora w przedstawiony powyżej sposób należy ocenić jakość regulacji w docelowym rozwiązaniu i przeprowadzić korekty ustawień regulatora, jeżeli zostaną zauważone przeregulowania lub oscylacje.

Korzystając z symulatora regulatora PID udostępnionego na stronie engineers-excel.com w formie pliku „PID_scrollbar.xls” [51], zostały przeprowadzone symulacje dla testowego obiektu, opisanego wzorem 4.6 i nastaw wyznaczonych przedstawionymi wyżej metodami. W pierwszej kolejności przeprowadzono symulację regulatora PI dla współczynników wyliczonych metodą Ziglera-Nicholsa.

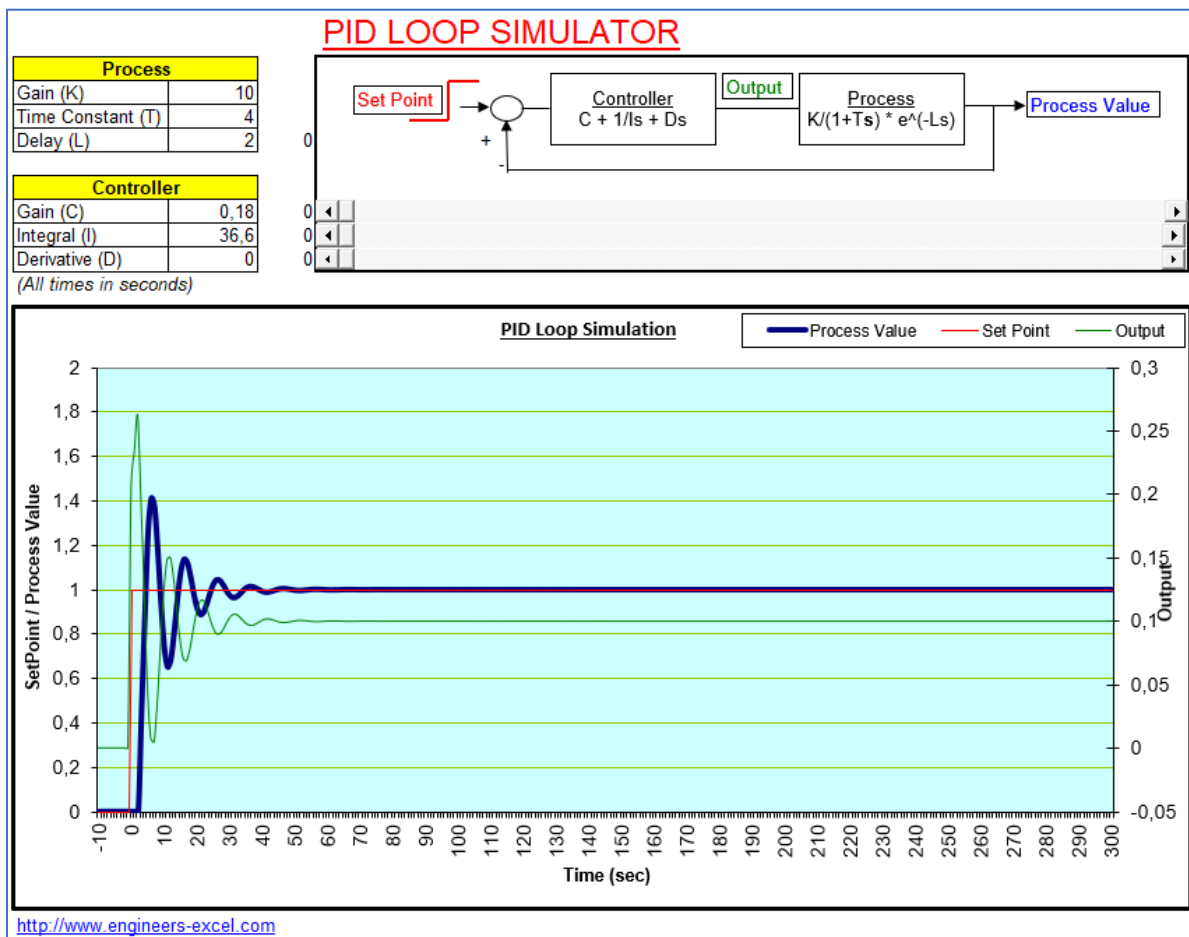
$$K_p = \frac{0,9 T_1}{K T_D} = \frac{0,9 * 4}{10 * 2} = 0,18 \quad (4.7)$$

$$T_i = 3,3 T_D = 3,3 * 2 = 6,6 \quad (4.8)$$

Następnie, uzyskane współczynniki możemy podstawić do funkcji regulatora PI:

$$C(s) = K_p \left(1 + \frac{1}{T_i s} \right) = 0,18 \left(1 + \frac{1}{4s} \right) = 0,18 + \frac{1}{36,6 s} \quad (4.9)$$

Jak widać na Rys. 4.43 wynikiem regulacji (Process Value) jest silna oscylacja gasnąca.



Rys. 4.43 Symulacja regulatora PI dla parametrów wyliczonych metodą Ziglera-Nicholsa [51]

W kolejnym kroku, dla:

$$\lambda = 1,9 T_1 = 1,9 * 4 = 7,6 \quad (4.10)$$

Zostały wyliczone parametry regulatora za pomocą metody Lambda:

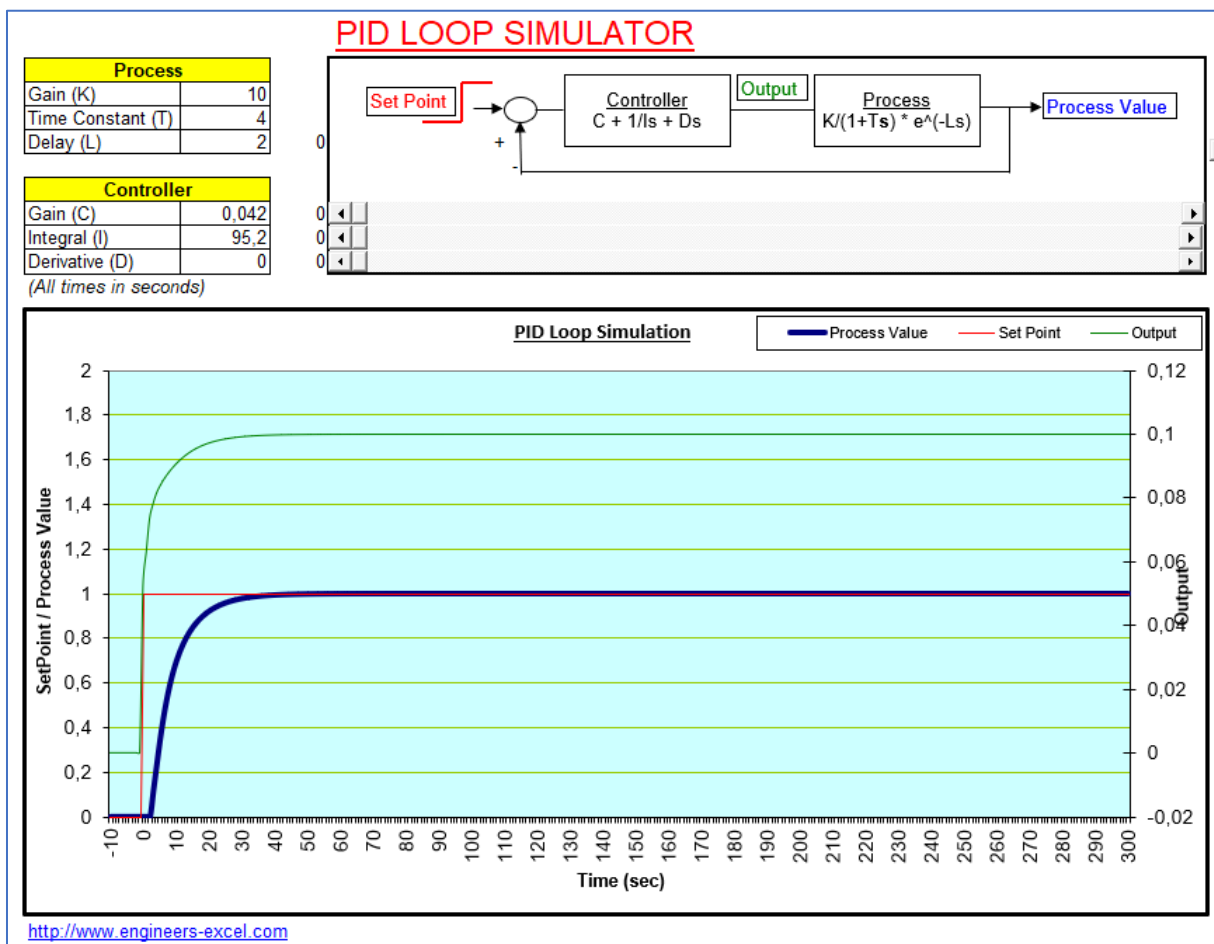
$$K_p = \frac{T_1}{K(\lambda + T_D)} = \frac{4}{10 * (7,6 + 2)} = 0,042 \quad (4.11)$$

$$T_i = T_1 = 4 \quad (4.12)$$

Po podstawieniu, możemy napisać:

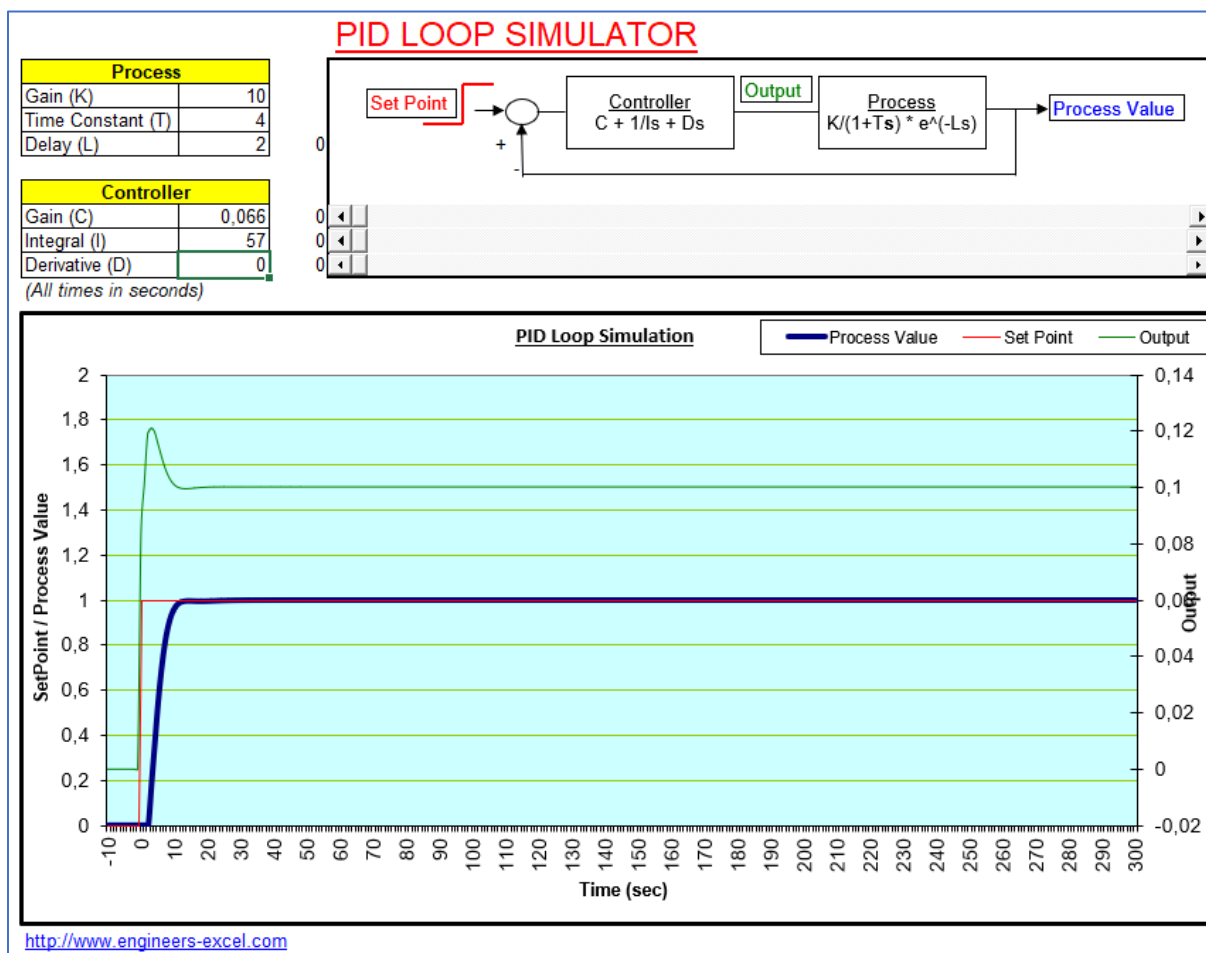
$$C(s) = K_p \left(1 + \frac{1}{T_i s} \right) = 0,042 \left(1 + \frac{1}{4s} \right) = 0,042 + \frac{1}{95,2 s} \quad (4.13)$$

Tak wyliczone parametry posłużyły do wygenerowania symulacji widocznej na Rys. 4.44. Jak widać poniżej, oscylacje zostały całkowicie wyeliminowane, jednak czas ustalania stanu ustalonego jest dość stosunkowo długi.



Rys. 4.44 Symulacja regulatora PI dla parametrów wyliczonych metodą Lambda [51]

Uzyskany poprzednio wynik można uznać za bardzo dobry, jednak symulacja komputerowa pozwala przeprowadzić ręczną modyfikację ustawień regulatora. Tą metodą, po kilku próbach, udało się wypracować kompromisową stabilizację, łączącą dużą szybkość stabilizacji z nieznacznym zafalowaniem, co zostało przedstawione na Rys. 4.45.

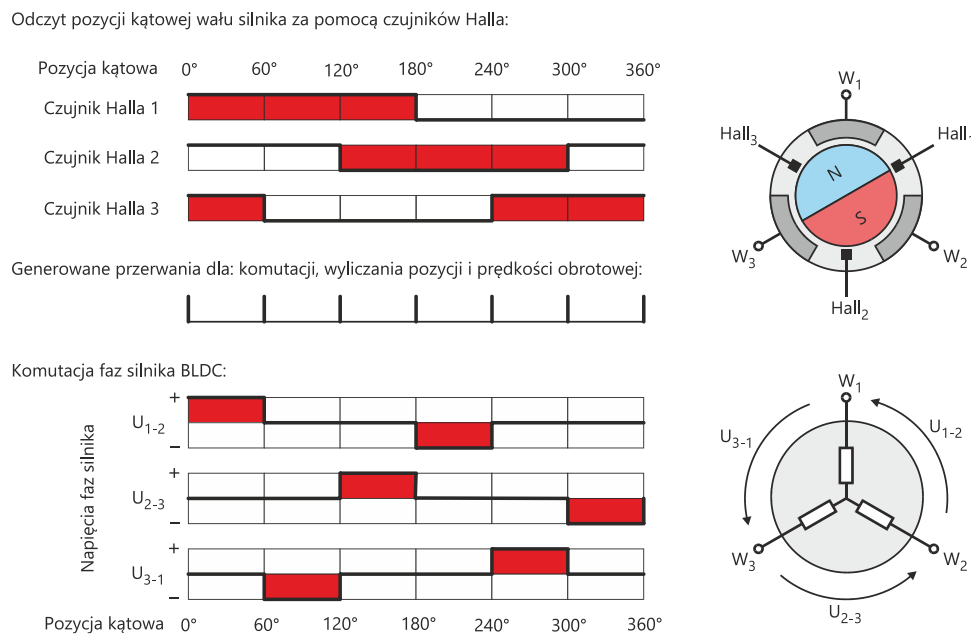


Rys. 4.45 Symulacja regulatora PI dla parametrów wyliczonych metodą Lambda z modyfikacją [51]

Komutacja faz silnika BLDC

Efektym wyjściowym regulatora PI jest za pomocą sygnału PWM i tranzystorów mocy, odpowiednie wysterowanie bezszczotkowego silnika prądu stałego z magnesami trwałymi. Taki silnik nazywany jest silnikiem BLDC (ang.: brushless direct-current motor). Silniki BLDC charakteryzują się wyższą sprawnością w stosunku do popularnych silników komutatorowych, a także zapewniają znacznie wyższą niezawodność z uwagi na wyeliminowanie zużywania się mechanicznego komutatora w postaci szczotek elektrycznych. Zastosowane w robocie silniki BLDC mają 3 uzwojenia fazowe połączone zazwyczaj w gwiazdę, co pozwala osiągnąć

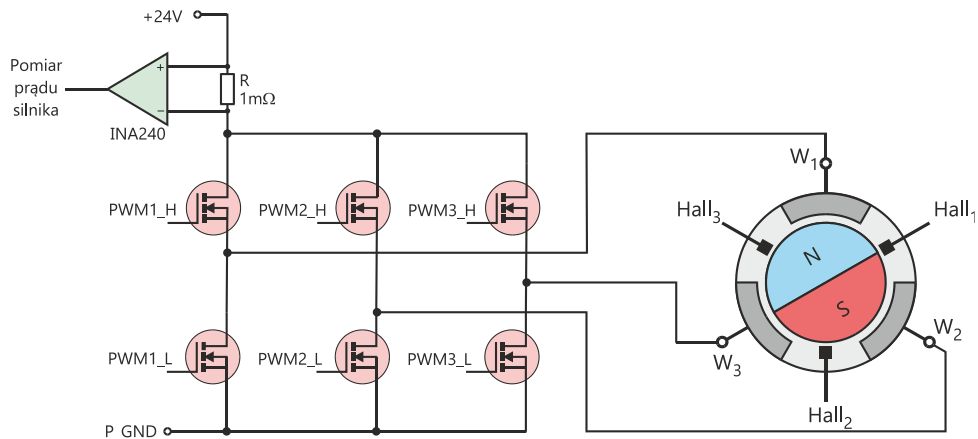
większą sprawność niż w przypadku połączenia w trójkąt [52]. Z uwagi na potrzebę pracy przy niskich prędkościach obrotowych, do budowy robota zastosowano silniki BLDC z wbudowanymi czujnikami Halla, które wykrywają zmianę biegunów magnesu. W danej chwili jest możliwa zmiana tylko na jednym czujniku. Warto też zauważyć, że silniki mogą mieć różną liczbę par biegunów, na rysunku Rys. 4.46 jest przedstawiony najprostszy wariant, jednej pary biegunów. Do budowy robota ARM-200 zastosowano również silniki firmy Maxon serii EC 60 Flat o siedmiu parach biegunów [53] i EC 45 Flat o ośmiu parach biegunów [40]. Sekwencja zmian biegunów odczytana za pomocą czujników Halla została przedstawiona na Rys. 4.46. Algorytm komutujący wykrywa i sygnalizuje stany niedozwolone, takie jak jednoczesny stan wysoki lub jednoczesny stan niski na trzech czujnikach Halla. Każda zmiana wyjścia czujników Halla generuje przerwanie, które podczas jego obsługi realizuje komutację faz silnika zgodnie z zadanym kierunkiem obrotu, a także inkrementowana/dekrementowana jest pozycja względna napędu i wyliczana jest, na podstawie pomiaru czasu pomiędzy impulsami, prędkość obrotowa. Komutacja faz silnika, przebiega zgodnie z sekwencją przedstawioną na Rys. 4.46.



Rys. 4.46 Opis sekwencji komutacji silnika BLDC z czujnikami Halla [54, s. 67]

Każda faza jest przełączana pomiędzy trzema stanami: podłączeniem do źródła zasilania (w tym wypadku 24V), podłączeniem do masy zasilania lub na pozostawieniu fazy niepodłączonej. W przypadku napięcia U₁₋₂ zaznaczonym na Rys. 4.46, w przedziale pozycji kątowej od 0° do 60° uzwojenie W₂ będzie podłączone do napięcia 24V, uzwojenie W₁ będzie

podłączone do masy zasilania a uzwojenie W3 pozostanie niepodłączone [54, s. 67]. Odwrócenie biegunów zasilania faz W1 i W2 wywoła zmianę kierunku wirowania wirnika silnika BLDC. W prezentowanym rozwiązaniu, mikrokontroler sterujący pracą modułu Slave wypracowuje 6 sygnałów: PWM1_H, PWM2_H, PWM3_H i PWM1_L, PWM2_L, PWM3_L, które sterują pracą tranzystorów mocy MOS-FET z kanałem N. Jak przedstawiono na Rys. 4.47, tranzystory pracują w tzw. pełnym mostku 3-fazowym, co pozwala na przyłączenie każdej z faz silnika do napięcia zasilania (24V) lub do potencjału masy P_GND, lub na pozostawienie danej fazy niepodłączonej. By umożliwić przepływ prądu i tym samym wywołanie momentu obrotowego silnia, wymagane jest podłączenie jednej fazy do potencjału dodatniego zasilania i innej fazy do potencjału masy. Nie jest dozwolone jednoczesne włączenie dwóch tranzystorów w jednym mostku, np. jednoczesne wystereowanie sygnału PWM1_H i PWM1_L, ponieważ prowadzi to do uszkodzenia tranzystorów mocy i/lub zasilania. Aby wyeliminować ryzyko wystąpienia tego zagrożenia, zastosowana sekwencja sterowania (Rys. 4.46) nie dopuszcza bezpośredniego przełączania pomiędzy sąsiadującymi stanami tej samej fazy z jednego stanu aktywnego do innego stanu aktywnego. Każde przełączenie poprzedzone jest wystąpieniem stanu odłączenia. Natomiast zwarcie wszystkich faz do potencjału zasilania (PWM1_H, PWM2_H, PWM3_H) lub do masy (PWM1_L, PWM2_L, PWM3_L) umożliwia realizację hamowania dynamicznego silnikiem.

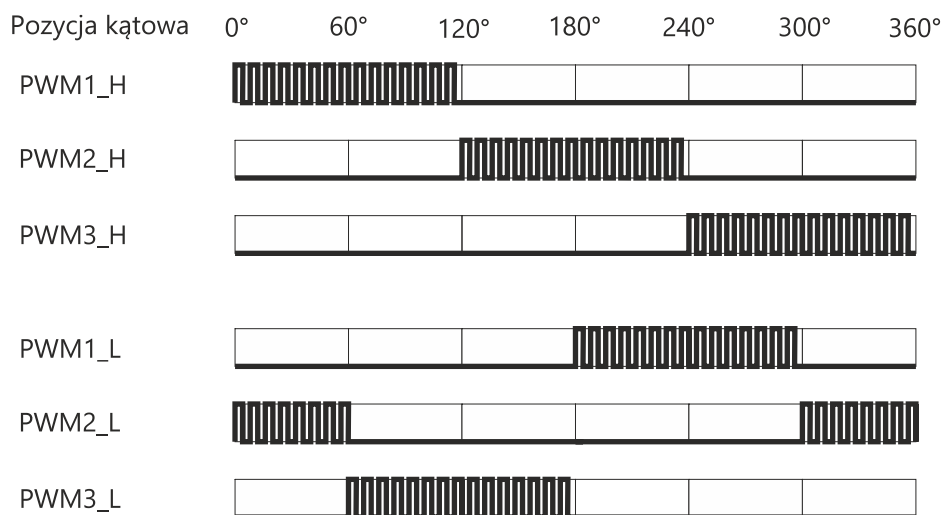


Rys. 4.47 Schemat sposobu sterowania fazami silnika BLDC

Pomiar prądu silnika, który może służyć do pośredniego pomiaru siły i wykrywania np. kolizji ramienia robota z przeszkodami (nie oddziałującymi na czujnik siły), został zrealizowany przez pomiar napięcia na rezystorze o bardzo małej rezystancji, o wartości 1mΩ, wpiętej szeregowo w obwód zasilania. Do pomiaru prądu użyto specjalizowanego układu firmy Texas Instruments INA240 [55], który umożliwia pomiar prądu na potencjale zasilania. Ułatwia to wylewanie

poligonów masy (GND), co może być istotne z punktu widzenia kompatybilności elektromagnetycznej. Dzięki pomiarowi prądu zasilania, program sterujący ma możliwość ograniczania prądu silnika, co jest ważne z punktu widzenia ochrony pacjenta, rehabilitanta obsługującego robota a także ochrony tranzystorów mocy i silnika przed uszkodzeniem.

Algorytm komutacji silnika, odpowiedzialny jest za właściwe przełączanie zasilania uzwojeń, co umożliwia wytworzenie momentu obrotowego w wybranym kierunku. W celu regulacji prędkości lub momentu obrotowego wymagana jest możliwość zmiany wypełnienia sygnałów PWM (ang.: Pulse Width Modulation) sterujących pracą tranzystorów mocy. W prezentowanym przykładzie, sygnał PWM jest przełączany ze stałą częstotliwością 20kHz, co zapewnia wyeliminowanie słyszalnego hałasu generowanego przez uzwojenia silnika. Wbudowany w mikrokontroler układ kontrolera generatorów PWM udostępnia 16-bitową rozdzielność ustawiania wypełnienia sygnału PWM, natomiast w projekcie użyta jest rozdzielczość 10-bitowa.

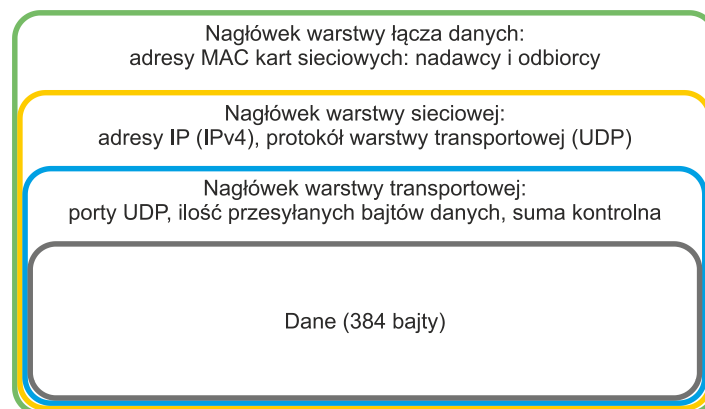


Rys. 4.48 Komutacja faz silnika BLDC z modulacją PWM

Jak widać na Rys. 4.48, sygnały komutujące fazy są modulowane sygnałem PWM z wykorzystaniem strategii kluczowania jednocześnie tranzystorów w górnej i dolnej części mostka [56, s. 51]. Takie rozwiązanie zapewnia równomierne obciążenie górnych i dolnych tranzystorów mocy a także uzyskanie krótszych czasów wyłączenia, co umożliwia zastosowanie wyższej częstotliwości sygnału PWM [57].

4.15. Protokół komunikacji

Istotnym elementem systemu sterowania robota jest łącze danych pomiędzy komputerem PC, a modulem Master systemu sterowania. Połączenie to zostało zrealizowane w warstwie fizycznej modelu OSI [58, s. 33] w postaci standardowego połączenia kablowego Ethernet 10/100 Mbps. Mikrokontroler SAM4E8C [59] modułu Master systemu sterowania, jest wyposażony w interfejs Ethernet 10/100 Mbps kompatybilny ze standardem IEEE 802.3, co w połączeniu z dostępnymi kanałami DMA i wsparciem sprzętowym mikrokontrolera podczas sprawdzania i liczenia sum CRC przesyłanych pakietów, znacznie ułatwia realizację responsywnego połączenia z komputerem PC. Warstwa sieciowa została oparta na protokole IPv4. Natomiast w warstwie transportowej modelu OSI został wykorzystany protokół internetowy UDP [60, s. 261–267]. Jest to protokół bezpołączeniowy, który nie ma narzutu związanego z nawiązywaniem połączenia i śledzenia sesji, co pozwala zmniejszyć opóźnienia w systemie sterowania robota. Pakiety danych, przesyłane w obu kierunkach, między robotem a aplikacją PC, po enkapsulacji niższymi warstwami stosu TCP/IP, zostały przedstawione w uproszczeniu na Rys. 4.49.



Rys. 4.49 Uproszczony schemat enkapsulacji przesyłanych danych

Bardziej szczegółowa budowa przesyłanych pakietów została opisana w Tab. 4.5, gdzie na najniższym poziomie (bajty 0-7) w warstwie łącza danych, typu Ethernet II, w preambule przesyłane są bity, na przemian „1” i „0”, synchronizujące obwody wejściowe karty sieciowej. Następnie, po ostatnich bitach „11” preambuły, przesyłane są: adresy sprzętowe MAC karty sieciowej nadawcy i karty sieciowej odbiorcy.

Tab. 4.5 Szczegółowy opis warstw i danych pakietu

	Bajty	0-7	8-15	16-23	24-31
Warstwa łącza danych	0 - 3	0b10101010	0b10101010	0b10101010	0b10101010
	4 - 7	0b10101010	0b10101010	0b10101010	0b10101011
	8 - 11	Adres MAC nadawcy			
	12 - 15	Adres MAC nadawcy		Adres MAC odbiorcy	
	16 - 19	Adres MAC odbiorcy			
	20 - 21	Typ: IPv4 (0x0800)			
Warstwa sieciowa	22 - 25	Adres IP nadawcy			
	26 - 29	Adres IP odbiorcy			
	30 - 33	0x00	Protokół: UDP (0x11)	Ilość bajtów datagramu UDP	
Warstwa transportowa	34 - 37	Port UDP nadawcy		Port UDP odbiorcy	
	38 - 41	Ilość bajtów danych		Suma kontrolna CRC	
	42 - 425	Dane (384 bajty)			
	426 - 429	Suma kontrolna ramki Ethernet CRC			

Kolejnym parametrem, przesyłanym w warstwie łącza danych, jest typ danych warstwy wyższej (warstwy sieciowej). W tym przypadku jest to znacznik 0x0800, sygnalizujący protokół IPv4. Dodatkowo, podczas enkapsulacji danych, przed wysłaniem kablem Ethernet, na koniec pakietu dołączana jest wyliczona 32-bitowa suma kontrolna CRC (bajty: 426-429). Dane transportowe pakietu zostały zaznaczone w Tab. 4.5 kolorem szarym. Przechodząc do warstwy sieciowej, tzw. pseudo nagłówka, zaznaczonego kolorem pomarańczowym, widzimy adresy logiczne IP (IPv4) urządzeń sieciowych nadane dynamicznie przez serwer DHCP danej sieci lub konfigurowane statycznie przez administratora. W tej warstwie, oprócz adresu IP nadawcy i odbiorcy, dostarczana jest również informacja o protokole w jakim przesyłane będą dane w warstwie wyższej (transportowej), w tym wypadku jest to bezpołączeniowy protokół UDP. Ostatnim elementem warstwy sieciowej jest długość datagramu UDP, na który składa się 8 bajtów nagłówka i 384 bajtów danych. Jak już zostało wcześniej wspomniane, w warstwie transportowej został zastosowany protokół bezpołączeniowy UDP (ang.: User Datagram Protocol), który charakteryzuje się szybkością. Szybkość tego protokołu wynika z braku

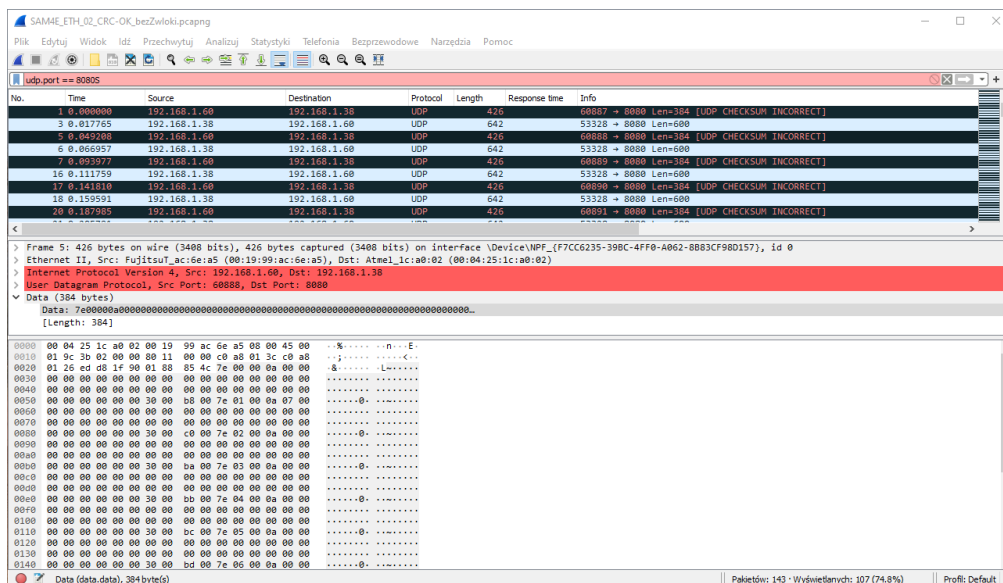
dodatkowych mechanizmów zabezpieczających, obecnych w alternatywnym protokole połączeniowym TCP (ang.: Transmission Control Protocol), który zapewnia np.: kontrolę sesji, potwierdzanie odbioru pakietu, czy retransmisję uszkodzonych pakietów. W naszym przypadku, z uwagi na cykliczne i częste przesyłanie danych (typu pytanie-odpowiedź), pozwalające zrealizować mechanizm kontroli utrzymania połączenia i dużą niewrażliwość na potencjalną utratę pojedynczych pakietów, wybrano szybszy protokół UDP [61, s. 142]. Nagłówek UDP składa się z portu nadawcy i z portu odbiorcy. Port UDP jest wartością 16-bitową, w której zakres 0–255 jest zarezerwowany dla usług standardowych np.: DNS, SNMP, TFTP. Natomiast porty wyższe, szczególnie w zakresie 1024–65535, mogą być już dość dowolnie wykorzystywane. W opisywanym projekcie, moduł Master nasłuchuje danych na porcie 8080. Na tym samym porcie, przez komputer PC, są również odbierane dane wysyłane z modułu Master do komputera PC. Kolejnym składnikiem nagłówka UDP jest 16-bitowa ilość bajtów, na którą składa się 8 bajtów nagłówka UDP oraz liczba przesyłanych danych. Ostatnim elementem nagłówka UDP jest suma CRC, zabezpieczająca przesyłane dane, która wyliczana jest z: pseudo nagłówka UDP zawierającego adresy IP warstwy sieciowej, nagłówka UDP i przesyłanych danych. Ilość przesyłanych danych jest uzależniona od liczby obsługiwanych węzłów. Opisywany system sterowania robota składa się z jednego modułu Master i modułów Slave, których liczba jest uzależniona od liczby obsługiwanych osi swobody z napędami. W przedstawionym robocie rehabilitacyjnym ARM–200 system sterowania obsługuje 7 modułów Slave. Ramki danych przekazywane z komputera PC do modułów wykonawczych Slave zostały opisane w Tab. 4.7. Natomiast w Tab. 4.8 została opisana ramka danych zwrotnych przesyłana od modułu Slave do aplikacji sterującej w komputerze PC. Jak widać, w obu kierunkach przesyłane jest 48 bajtów danych dla każdego węzła. Z tych ramek budowana jest ramka danych UDP zawierająca dane modułu Master (adres 0) i dane modułów Slave (adresy 1–7). Struktura danych pakietu UDP została opisana w Tab. 4.6. W sumie, w jednym pakiecie UDP, przesyłane jest 384 bajtów danych, które sterują pracą wszystkich węzłów wykonawczych. Przesyłanie tak małej ilości danych dodatkowo przyspiesza proces komunikacji, eliminując problem fragmentacji i późniejszej defragmentacji odebranych danych. Teoretycznie, w pakiecie UDP istnieje możliwość przesłania 64kB danych z uwagi na 16-bitowe pole liczby danych, jednak w praktyce MTU (ang.: Maximum Transmission Unit) urządzeń sieciowych, czyli maksymalna porcja danych jaką można przesłać bez uruchamiania mechanizmu fragmentacji, jest ograniczona do 1500 bajtów. Przyjmując to ograniczenie,

można założyć, że zaproponowana struktura ramek pozwala na obsługę maksymalnie aż 31 węzłów wykonawczych.

Tab. 4.6 Ogólny opis struktury danych pakietu UDP robota ARM-200

Bajty	Adres	Opis ramki systemu sterowania
0-47	0	Obsługa modułu Master
48-96	1	Obsługa modułu Slave
97-144	2	Obsługa modułu Slave
145-192	3	Obsługa modułu Slave
193-240	4	Obsługa modułu Slave
241-288	5	Obsługa modułu Slave
289-336	6	Obsługa modułu Slave
337-384	7	Obsługa modułu Slave

W celach testowych, do kontroli przesyłanych pakietów pomiędzy komputerem PC a modulem Master, została użyta aplikacja analizatora protokołów Wireshark [62], która dzięki możliwości filtrowania pakietów i sygnalizacji błędnych pakietów znacząco ułatwiła uruchomienie komunikacji sieciowej systemu sterowania. Rys. 4.50 przedstawia okno programu Wireshark z przechwyconym przykładowym pakietem UDP. Dane pomiędzy komputerem PC, a modulem Master przesyłane są w pakietach danych o stałej długości 384 bajtów.



Rys. 4.50 Przykład przesyłanego pakietu UDP w oknie programu Wireshark

W ramach realizacji projektu robota rehabilitacyjnego ARM-200, został opracowany dedykowany protokół wymiany danych pomiędzy aplikacją sterującą, a węzłami robota. Wymiana informacji pomiędzy elementami robota odbywa się tylko za pomocą dwóch typów ramek, „Ramki Rozkazu” i „Ramki Odpowiedzi”. „Ramka Rozkazu” opisana w Tab. 4.7, przesyła informacje z aplikacji sterującej uruchomionej na komputerze PC do modułu Master lub modułów Slave za pośrednictwem modułu Master. Natomiast „Ramka Odpowiedzi”, opisana w Tab. 4.8, przesyła dane zwrotne zawierające stan i parametry pracy węzłów wykonawczych. „Ramka Rozkazu” składa się z charakterystycznego nagłówka (bajt o wartości 0x7E), następnie przesyłany jest adres węzła wykonawczego, do którego kierowane są dane. Adres 0 zarezerwowany jest dla komunikacji z modułem Master, adresy 1–7 służą do adresowania modułów Slave. Kolejnym elementem ramki jest 8-bitowy licznik ramek, który pozwala wykryć utratę ramki lub zapętlenie wysyłania tej samej ramki. W kolejnym bajcie przesyłany jest typ przekazywanych danych (**data_c_type**). Za jego pomocą strona odbiorcza informowana jest o przeznaczeniu danych w 40-bajtowym polu danych (**data_cmd**). Opis obecnie zaimplementowanych typów danych (**data_c_type**) został przedstawiony w Tab. 4.9, a opracowany protokół pozwala również na dodanie w przyszłości innych typów danych. Ostatnie dwa bajty „Ramki rozkazów” są to bajty sumy kontrolnej CRC, które umożliwiają wykrycie przekłamania przesyłanych danych. Tę sumę CRC weryfikuje tylko węzeł wykonawczy, do którego jest zaadresowany pakiet.

Tab. 4.7 Ramka rozkazów z PC do modułu Slave

Bajt	Parametr	Typ	Opis
0	header	uint8	Nagłówek pakietu danych (0x7E)
1	node_adr	uint8	Adres węzła Master (0), Slave (1-7)
2	pakiet_cnt	uint8	Licznik wysłanych pakietów
3	data_c_type	uint8	Typ przesyłanych danych w polu data_cmd , opis w Tab. 4.9
4 - 44	data_cmd	40x uint8	Dane zapytania / rozkazu zależne od poprzedniego parametru data_c_type
45	res	int16	wolne
46	res		
47	crc	int16	Suma kontrolna CRC-16
48	crc		

W odpowiedzi na otrzymanie „Ramki Rozkazu” węzeł wykonawczy odsyła „Ramkę odpowiedzi”, opisaną w Tab. 4.8, która jest zbudowana w podobny sposób jak „Ramka Rozkazu”. Zmieniony został nagłówek, na wartość 0xAA i lista typów danych przesyłanych w odpowiedzi. Opis możliwych typów danych został przedstawiony w Tab. 4.10. Kontrola stanu połączenia aplikacji sterującej z robotem jest realizowana równoległe podczas cyklicznej wymiany danych. Brak odbioru „Ramek odpowiedzi”, w czasie pomiędzy kolejnymi wysłanymi „Ramkami Rozkazów” jest wykrywany i sygnalizowany w aplikacji sterującej, co po przekroczeniu zadanej ilości braków odpowiedzi, jest sygnalizowane jako utrata połączenia z robotem. Niezależnie, moduł Master robota oczekuje odbioru kolejnych pakietów „Ramki Rozkazów”. Jeżeli zostanie przekroczony ustawiony czas na odbiór pakietu z aplikacji sterującej, który wynosi 4-krotność ustawionego czasu pomiędzy standardowymi przesłaniami pakietu, robot przechodzi do stanu zatrzymania, w którym blokowane są napędy hamulcami i wymagane jest ponowne przesłanie rozkazu startu pracy robota.

Tab. 4.8 Ramka odpowiedzi z Slave do PC przesyłane są w postaci ramek danych

Bajt	Parametr	Typ	Opis
0	header	uint8	Nagłówek pakietu danych (0xAA)
1	node_adr	uint8	Adres węzła Master (0), Slave (1-7)
2	pakiet_cnt	uint8	Kontrolny licznik pakietów, kopiowany z zapytania
3	data_r_type	uint8	Typ przesyłanych danych w polu data_replay , opis w
4 - 44	data_replay	40x uint8	Dane odpowiedzi zależne od poprzedniego parametru data_r_type
45	res	int16	wolne
46	res		
47	crc	int16	Suma kontrolna CRC-16
48	crc		

Podczas standardowej pracy robota w „Ramce Rozkazu” parametr **data_c_type** sygnalizuje przesyłanie pakietów **Control** (0xA1), które przekazują zestaw informacji pozwalający na ustawienie trybów pracy węzła (Tab. 4.16), sposobu pomiaru siły nacisku (Tab. 4.17) i wartości parametrów określających prędkość, a także zadaną pozycję oraz inne informacje wymagane do właściwej pracy węzła. Opis tych parametrów został umieszczony w Tab. 4.11. Moduły Slave, przed przystąpieniem do właściwej pracy muszą zostać

odpowiednio skonfigurowane, za pomocą przesłanych im pakietów konfiguracyjnych: **Config1** (0xA2), **Config2** (0xA3), **Config3** (0xA4). Jeżeli dany moduł Slave nie otrzyma wszystkich wymaganych pakietów konfiguracyjnych nie będzie możliwe przejście do normalnej pracy. Brak skonfigurowania modułu Slave będzie sygnalizowany brakiem ustawienia jednego lub kilku bitów 0-2 w statusie **NodeWarnings** modułu Slave, opis w Tab. 4.19.

Tab. 4.9 Lista typów danych *data_c_type* rozkazów przesyłanych z PC do modułów

Kod pakietu	Nazwa pakietu
0xA1	Pakiet Control (tryb podstawowy), opis w Tab. 4.11
0xA2	Pakiet Config1 – Ustawienie konfiguracji 1 dla węzła, opis w Tab. 4.12
0xA3	Pakiet Config2 – Ustawienie konfiguracji 2 dla węzła, opis w Tab. 4.13
0xA4	Pakiet Config3 – Ustawienie konfiguracji 2 dla węzła, opis w Tab. 4.14
0xA5	Pakiet FirmwareVersion – Odczyt numerów wersji programów wbudowanych, opis w Tab. 4.15
0xA6	Pakiet FirmwareUploadCmd – Ustawienie przejścia w tryb aktualizacji (do przyszłej implementacji)
0xA7	Pakiet FirmwareUploadData – Przesyłanie danych aktualizacji (do przyszłej implementacji)
0xA8	Pakiet Config1Get – Odczyt konfiguracji 1 dla węzła Tab. 4.12
0xA9	Pakiet Config2Get – Odczyt konfiguracji 2 dla węzła
0xAA	Pakiet Config3Get – Odczyt konfiguracji 3 dla węzła

Z uwagi na duże znaczenie konfiguracji danego węzła wykonawczego, opracowany protokół umożliwia odczytanie przesłanych parametrów konfiguracyjnych w celu weryfikacji ich poprawności za pomocą rozkazów: **Config1Get** (0xA8), **Config2Get** (0xA9), **Config3Get** (0xAA). Dodatkowo, protokół umożliwia odczytanie wersji oprogramowania za pomocą rozkazu **FirmwareVersion** (0xA5), co jest szczególnie istotne z uwagi na wymaganie identyfikacji oprogramowania w wyrobie medycznym. Przewidziano również w protokole możliwość aktualizacji oprogramowania wbudowanego węzłów wykonawczych, za pomocą dwóch rozkazów: **FirmwareUploadCmd** (0xA6) i **FirmwareUploadData** (0xA7). Pierwszy rozkaz (**FirmwareUploadCmd**) służy do uwierzytelnienia i przełączenia w tryb aktualizacji a drugi (**FirmwareUploadData**) sygnalizuje przekazanie kolejnej porcji danych pliku wsadowego. Obecnie, ta funkcjonalność nie jest zaimplementowana.

Tab. 4.10 Lista typów danych data_r_type odpowiedzi przesyłanych z modułów do PC

Kod pakietu	Nazwa pakietu
0xB1	Pakiet ControlReplay (tryb podstawowy), opis w Tab. 4.20
0xB2	Pakiet Config1Replay – Odpowiedz na Config1Get, opis w Tab. 4.21
0xB3	Pakiet Config2Replay – Odpowiedz na Config2Get, opis w Tab. 4.22
0xB4	Pakiet Config3Replay – Odpowiedz na Config3Get, opis w Tab. 4.23
0xB5	Pakiet FirmwareVersionReplay
0xB6	Pakiet FirmwareUploadCmdReplay
0xB7	Pakiet FirmwareUploadDataReplay
0xC1	Pakiet ControlMasterReplay (odpowiedz tylko węzła MASTER (adr:0))

W tabeli powyżej (Tab. 4.10) opisane są możliwe odpowiedzi węzłów wykonawczych. Podczas normalnej pracy robota, po nawiązaniu połączenia i przesłaniu pakietów konfiguracyjnych węzły systemu, moduły Slave w reakcji na cyklicznie odbierany pakiet **Control** (0xA1), odsyłają pakiet **ControlReplay** (0xB1) opisany w Tab. 4.20, natomiast moduł Master odsyła pakiet **ControlMasterReplay** (0xC1), którego opis został umieszczony w Tab. 4.25. Jak widać w Tab. 4.11, pakiet **Control** (0xA1) przekazuje podstawowe informacje o oczekiwanym trybie pracy i parametrach sterujących, pozwala to dynamicznie przełączać zachowanie węzłów robota w zależności od wybranego scenariusza pracy.

Tab. 4.11 Opis pakietu Control (0xA1)

Bajt	Nazwa parametru (numer bajtu)	Typ	Opis	Uwagi
0	mode (0)	uint32	Tryb pracy węzła: bity 3-0, opis w Tab. 4.16	
1	mode (1)		Tryb pomiaru siły: bity 6-4, opis w Tab. 4.17	
2	mode (2)			
3	mode (3)			
4	force_max (0)	uint16	Ustawienie maksymalnej siły nacisku (dla trybu odtwarzania trajektorii)	Rozdzielczość: 0,1[N] Zakres: 100,0[N]
5	force_max (1)			
6	force_insens (0)	uint16	Ustawienie progu nieczułości siły nacisku (dla trybu śledzenia trajektorii)	Rozdzielczość: 0,1[N] Zakres: 100,0[N]
7	force_insens (1)			
8	position_set (0)	int32	Ustawienie nowej pozycji (dla trybu odtwarzania trajektorii)	Rozdzielczość: 0.1[°] Zakres: ±180[°]
9	position_set (1)			
10	position_set (2)			

11	position_set (3)			
12	speed_set (0)	uint16	Ustawienie prędkości przejścia do nowej pozycji (dla trybu odtwarzania trajektorii)	Rozdzielczość: 0.1[obr/min]
13	speed_set (1)			Zakres: 100.0[obr/min]
14	pwm_set (0)	int16	Ustalenie nowego wypełnienia PWM (dla trybu MODE_PWM)	Rozdzielczość: 1
15	pwm_set (1)			Zakres: ±1000
16	actuator_set (0)	uint16	Ustalenie rozsunienia siłownika (dla trybu MODE_OFF_LENGTH_SET)	Rozdzielczość: 1[mm]
17	actuator_set (1)			Zakres: 0-100[mm]
18				
...				
39				

Warto zwrócić uwagę, że pakiet konfigurujący węzeł **Config1** (0xA2), opisany w Tab. 4.12, przesyła niezależnie informacje o ograniczeniu zakresu kąтового ruchu robota dla strony prawej i lewej. Wynika to z faktu, że konstrukcja mechaniczna robota ARM-200 nie jest symetryczna, co wymaga indywidualnego ograniczenia dla każdej konfiguracji. Warto również zwrócić uwagę, że każdy z pakietów konfiguracyjnych, na bajtach 0-3 przesyła informacje o trybie pracy robota, takie rozwiązanie pozwala szybko zmienić tryb pracy, np. zatrzymać robota (w wypadku przesyłania ponownie pakietów konfiguracyjnych np. podczas prac rozwojowych systemu).

Tab. 4.12 Opis pakietu Config1 (0xA2)

Bajt	Nazwa parametru (numer bajtu)	Typ	Opis	Uwagi
0	mode (0)	uint32	Tryb pracy węzła: bity 3-0, opis w Tab. 4.16	
1	mode (1)		Tryb pomiaru siły: bity 6-4, opis w Tab. 4.17	
2	mode (2)			
3	mode (3)			
4	position_min_right (0)	int32	Ustawienie ograniczenia pozycji (kąta) MIN dla strony PRAWYJ	Rozdzielczość: 0.1[°] Zakres: ±180[°]
5	position_min_right (1)			
6	position_min_right (2)			
7	position_min_right (3)			
8	position_max_right (0)	int32	Ustawienie ograniczenia pozycji (kąta) MIN dla strony PRAWYJ	Rozdzielczość: 0.1[°] Zakres: ±180[°]
9	position_max_right (1)			
10	position_max_right (2)			
11	position_max_right (3)			
12	position_min_left (0)	int32	Ustawienie ograniczenia pozycji (kąta) MIN dla strony LEWEJ	Rozdzielczość: 0.1[°] Zakres: ±180[°]
13	position_min_left (1)			
14	position_min_left (2)			

15	position_min_left (3)			
16	position_max_left (0)	int32	Ustawienie ograniczenia pozycji (kąta) MIN dla strony LEWEJ	Rozdzielczość: 0.1[°] Zakres: ±180[°]
17	position_max_left (1)			
18	position_max_left (2)			
19	position_max_left (3)			
20	speed_max (0)	uint16	Ustawienie ograniczenia maksymalnej prędkości	Rozdzielczość: 0.1[obr/min] Zakres: 100.0[obr/min]
21	speed_max (1)			
22	speed_rampa_up (0)	uint16	Ustawienie szybkości narastania prędkości	Rozdzielczość: 0.1[rpm/s] Zakres: 100[rpm/s]
23	speed_rampa_up (1)			
24	speed_rampa_down (0)	uint16	Ustawienie szybkości opadania prędkości	Rozdzielczość: 0.1[rpm/s] Zakres: 100[rpm/s]
25	speed_rampa_down (1)			
26	pwm_max (0)	uint16	Ustawienie ograniczenia PWM silnika	Rozdzielczość: 1 Zakres: 1000
27	pwm_max (1)			
28	current_max_pulse (0)	uint16	Ustawienie ograniczenia prądu chwilowego silnika (brak implementacji)	Rozdzielczość: 1 Zakres: 60000 [mA]
29	current_max_pulse (1)			
30	current_max_pulse_time (0)	uint16	Ustawienie czasu ograniczenia prądu chwilowego silnika (brak implementacji)	Rozdzielczość: 1 [ms] Zakres: 3000 [ms]
31	current_max_pulse_time (1)			
32	current_max_cont (0)	uint16	Ustawienie ograniczenia prądu ciągłego silnika	Rozdzielczość: 1[mA] Zakres: 60000 [mA]
33	current_max_cont (1)			
34	voltage_min (0)	uint16	Ustawienie minimalnego napięcia zasilania silnika	Rozdzielczość: 1[mV] Zakres: 60000 [mV]
35	voltage_min (1)			
36	pwm_shift (0)	uint16	Ustawienie minimalnego wypełnienia PWM	Rozdzielczość: 1 Zakres: 1000
37	pwm_shift(1)			
38				
39				

Tab. 4.13 Opis pakietu Config2 (0xA3)

Bajt	Nazwa parametru (numer bajtu)	Typ	Opis	Uwagi
0	mode (0)	uint32	Tryb pracy węzła: bity 3-0, opis w Tab. 4.16 Tryb pomiaru siły: bity 6-4, opis w Tab. 4.17	
1	mode (1)			
2	mode (2)			
3	mode (3)			
4	pid_speed_p (0)	uint32	Ustawienie regulatora PID – prędkość parametr P	Rozdzielczość: 0,01 Zakres: 1000
5	pid_speed_p (1)			
6	pid_speed_p (2)			
7	pid_speed_p (3)			
8	pid_speed_i (0)	uint32		Rozdzielczość: 0,01

9	pid_speed_i (1)		Ustawienie regulatora PID – prędkość parametr I	Zakres: 1000
10	pid_speed_i (2)			
11	pid_speed_i (3)			
12	pid_speed_d (0)	uint32	Ustawienie regulatora PID – prędkość parametr D	Rozdzielczość: 0,01 Zakres: 1000
13	pid_speed_d (1)			
14	pid_speed_d (2)			
15	pid_speed_d (3)			
16	pid_speed_t (0)	uint32	Ustawienie regulatora PID – prędkość parametr T	Rozdzielczość: 1[ms] Zakres: 1000
17	pid_speed_t (1)			
18	pid_speed_t (2)			
19	pid_speed_t (3)			
20	pid_position_p (0)	uint32	Ustawienie regulatora PID – pozycja parametr P	Rozdzielczość: 0,01 Zakres: 1000
21	pid_position_p (1)			
22	pid_position_p (2)			
23	pid_position_p (3)			
24	pid_position_i (0)	uint32	Ustawienie regulatora PID – pozycja parametr I	Rozdzielczość: 0,01 Zakres: 1000
25	pid_position_i (1)			
26	pid_position_i (2)			
27	pid_position_i (3)			
28	pid_position_d (0)	uint32	Ustawienie regulatora PID – pozycja parametr D	Rozdzielczość: 0,01 Zakres: 1000
29	pid_position_d (1)			
30	pid_position_d (2)			
31	pid_position_d (3)			
32	pid_position_t (0)	uint32	Ustawienie regulatora PID – pozycja parametr T	Rozdzielczość: 1[ms] Zakres: 1000
33	pid_position_t (1)			
34	pid_position_t (2)			
35	pid_position_t (3)			
36				
...				
39				

Tab. 4.14 Opis pakietu Config3 (0xA3)

Bajt	Nazwa parametru (numer bajtu)	Typ	Opis	Uwagi
0	mode (0)	uint32	Tryb pracy węzła: bity 3-0, opis w Tab. 4.16	
1	mode (1)		Tryb pomiaru siły: bity 6-4, opis w Tab. 4.17	
2	mode (2)			
3	mode (3)			
4	temp_driver_max (0)	uint16	Ustawienie alarmu przekroczenia temperatury tranzystorów mocy	Rozdzielczość: 0,1[°C]
5	temp_driver_max (1)			Zakres: 0 - 150,0[°C]

6	temp_motor_max (2)	uint16	Ustawienie alarmu przekroczenia temperatury silnika (opcja)	Rozdzielczość: 0,1[°C]
7	temp_motor_max (3)			Zakres: 0 - 150,0[°C]
8	position_h_gear_ratio (0)	uint32	Ustawienie ilości impulsów z czujników Hall'a na pełny obrót wyjścia przekładni (przełożenie przekładni * ilość biegunów silnika * 6)	Rozdzielczość: 1 Zakres: 60000
9	position_h_gear_ratio (1)			
10	position_h_gear_ratio (2)			
11	position_h_gear_ratio (3)			
12	position_e_gear_ratio (0)	uint32	Ustawienie ilości impulsów z enkodera silnika na pełny obrót wyjścia przekładni (przełożenie przekładni * ilość impulsów na obrót enkodera) (opcja)	Rozdzielczość: 1 Zakres: 60000
13	position_e_gear_ratio (1)			
14	position_e_gear_ratio (2)			
15	position_e_gear_ratio (3)			
16	node_cfg_bits (0)	uint32	Ustawienie konfiguracji węzła	bit0 -negacja pozycji a kątowej bit1 -negacja pozycji h kątowej bit2 -negacja pozycji e kątowej bit8 -podłączony hamulec bit9 -podłączony enkoder silnika bit10 -podłączony siłownik bit11 -podłączony enkoder liniowy bit12 -podłączony czujnik temperatury silnika
17	node_cfg_bits (1)			
18	node_cfg_bits (2)			
19	node_cfg_bits (3)			
20	force_gain (0)	uint32	Ustawienie wzmocnienia kanału pomiarowego siły	Rozdzielczość: 1 Zakres: 1000
21	force_gain (1)			
22	force_gain (2)			
23	force_gain (3)			
24	emg_sample_rate (0)	uint16	Ustawienie częstotliwości próbkowania sygnału EMG	Rozdzielczość: 1 Zakres: 125,250,500,1000
25	emg_sample_rate (1)			
26	emg_ch1_gain (0)	uint8	Ustawienie wzmocnienia EMG dla kanału 1	Rozdzielczość: 1 Zakres: 1,2,3,4,5,6,12
27	emg_ch2_gain (0)	uint8	Ustawienie wzmocnienia EMG dla kanału 2	Rozdzielczość: 1 Zakres: 1,2,3,4,5,6,12
28	emg_avg_time (0)	uint8	Ustawienie rozmiaru okna uśredniania sygnału	Rozdzielczość: 1[ms] Zakres: 1 – 100 [ms]
29	force_coff (0)	uint16	Współczynnik do wyliczania pomiaru siły z belki tensometrycznej	Dla belki 100N : 87 .
30	force_coff (1)			Dla belki 200N : 173 .

31				
...				
39				

Tab. 4.15 Opis pakietu *FirmwareVersion* (0xA5)

Bajt	Nazwa parametru (numer bajtu)	Typ	Opis	Uwagi
0	mode (0)	uint32	Tryb pracy węzła: bity 3-0, opis w Tab. 4.16	
1	mode (1)		Tryb pomiaru siły: bity 6-4, opis w Tab. 4.17	
2	mode (2)			
3	mode (3)			
4				
...				
39				

Tab. 4.16 prezentuje możliwe tryby pracy modułu Slave, jak widać jest to dość bogata lista, choć większość trybów służy głównie do celów testowych, kalibracyjnych lub jest przydatna podczas dopasowywania robota do danego pacjenta. Tutaj również warto zwrócić uwagę na niezależne rozkazy zerowania bezwzględnych czujników pozycji kątowej: `MODE_CAL_POSITION_ABS_R` i `MODE_CAL_POSITION_ABS_L`. Wynika to z obserwacji, że w zależności od konfiguracji robota, na stronę lewą lub prawą, luzy w niektórych przekładniach wybierane były w przeciwnych kierunkach co uniemożliwiało precyzyjne zapamiętanie pozycji zerowej dla lewej i prawej strony.

Tab. 4.16 Tryby pracy modułu *Slave*

nr	NodeMode Bity 3-0	Tryb pracy	Opis trybu
0.	0000	MODE_OFF	Wyłączenie napędu i siłownika, hamulec blokuje ruch (tryb podstawowy – wyłączenie pracy, stan bezpieczny)
1.	0001	MODE_OFF_LENGTH_MAN	Wyłączenie napędu głównego, możliwość ręcznego sterowania siłownikiem zmiany długości za pomocą przycisków na robocie
2.	0010	MODE_OFF_LENGTH_SET	Wyłączenie napędu głównego, możliwość sterowania rozsunieniem siłownika zmiany długości z systemu
3.	0011	MODE_POSITION	Sterowanie napędem głównym przez zadanie pozycji (tryb podstawowy - trzymanie pozycji / odtwarzanie ruchu - ćwiczenia pasywne)
4.	0100	MODE_FORCE_POSITION	Sterowanie napędem głównym przez czujnik siły, który steruje prędkością (tryb podstawowy - uczenie trajektorii - ćwiczenia aktywne)
5.	0101	MODE_CAL_FORCE	Zerowanie kanału pomiarowego siły

6.	0110	MODE_CAL_POSITION_REL	Zerowanie pozycji kątowej czujników względnych
7.	0111	MODE_CAL_POSITION_ABS_R	Zerowanie pozycji kątowej czujników bezwzględnych dla strony Prawej i zapamiętanie w pamięci EEPROM
8.	1000	MODE_CAL_POSITION_ABS_L	Zerowanie pozycji kątowej czujników bezwzględnych dla strony Lewej i zapamiętanie w pamięci EEPROM
9.	1001	MODE_PWM	Sterowanie napędem głównym przez zadanie PWM (tylko do testów!!!)
10.	1010	MODE_SPEED	Sterowanie napędem głównym przez zadanie prędkości (tylko do testów!!!)
11.	1011	MODE_FORCE_PWM	Sterowanie napędem głównym przez czujnik siły, który steruje PWM (tylko do testów!!!)
12.	1100	MODE_POSITION_NLIM	Sterowanie napędem głównym przez zadanie pozycji z rozszerzonym zakresem ruchu $\pm 270^\circ$ (tylko do przestawiania pozycji do HOME LEFT/RIGHT, z ograniczeniem prędkości)
13.	1101	MODE_FORCE_POSITION_NLIM	Sterowanie napędem głównym przez czujnik siły, który steruje prędkością bez ograniczenia zakresu (tylko do ustawiania pozycji przed kalibracją: MODE_CAL_POSITION_ABS_R i L)
14.	1110	MODE_SET_SIDE_LEFT	Przestawienie robota na stronę LEFT, zapamiętanie w EEPROM i synchronizacja z MODE_CAL_POSITION_ABS_R (przestawienie liczenia katów, po wcześniejszym obróceniu ramienia za pomocą MODE_POSITION_NLIM)
15.	1111	MODE_SET_SIDE_RIGHT	Przestawienie robota na stronę RIGHT, zapamiętanie w EEPROM i synchronizacja z MODE_CAL_POSITION_ABS_R (przestawienie liczenia katów, po wcześniejszym obróceniu ramienia za pomocą MODE_POSITION_NLIM)

W Tab. 4.17 przedstawione zostały możliwe konfiguracje toru pomiarowego siły. Jak już zostało wspomniane, moduł Slave posiada dwa kanały pomiaru siły, za pomocą konfiguracji bitów 6-4 trybu pracy węzła, możliwa jest praca z wejściem 1 lub 2, z pomiarem reagującym na sumę lub różnicę obu wejść. Możliwe jest także zanegowanie dowolnej konfiguracji.

Tab. 4.17 Tryby pomiaru siły modułu Slave

nr	NodeMode Bity 6-4	Tryb pracy	Opis trybu
0.	000	INPUT1	Pomiar z wejścia 1
1.	001	INPUT1_M	Pomiar z wejścia 1
2.	010	INPUT2	Pomiar z wejścia 2
3.	011	INPUT2_M	Pomiar z wejścia 2

4.	100	INPUT1_M_INPUT2	Pomiar różnicowy z wejścia 1 i wejścia 2
5.	101	INPUT2_M_INPUT1	Pomiar różnicowy z wejścia 2 i wejścia 1
6.	110	INPUT1_P_INPUT2	Pomiar sumacyjny z wejścia 1 i wejścia 2
7.	111	INPUT1_P_INPUT2_M	Pomiar sumacyjny z wejścia 1 i wejścia 2 znak minus

Każdy z węzłów Slave zapamiętuje w pamięci nieulotnej (EEPROM), poza pozycjami zerowymi dla każdej konfiguracji robota, także ustawioną wcześniej i zapamiętaną konfigurację (na stronę lewą lub prawą). Pozwala to wybrać właściwe, dla danej strony, kątowe zakresy ruchu robota. Przedstawione bity 9-8 w Tab. 4.18 umożliwiają odczytanie ustawionej strony robota lub stanu uniemożliwiającego dalszą pracę, jak brak ustawienia (pamięć EEPROM jeszcze nie zaprogramowana) lub uszkodzenie pamięci EEPROM.

Tab. 4.18 Strona pracy modułu Slave

nr	NodeMode Bity 9-8	Strona pracy robota	Opis trybu
0.	00	SIDE_NOT_SET	Błąd, brak ustawienia
1.	01	SIDE_LEFT	Węzeł robota ustawiony na stronę LEWA
2.	10	SIDE_RIGHT	Węzeł Robota ustawiony na stronę PRAWA
3.	11	SIDE_ERROR	Błąd pamięci EEPROM

W Tab. 4.19 zostały przedstawione flagi statusowe i flagi błędów modułu Slave, pozwala to w prosty sposób analizować pracę węzła a ta także diagnozować przyczyny awarii.

Tab. 4.19 Bity statusu NodeWarnings modułu Slave

nr bitu	Opis flagi
0	Moduł przeszedł pomyślnie konfiguracje1, odebrano konfiguracje Config1
1	Moduł przeszedł pomyślnie konfiguracje2, odebrano konfiguracje Config2
2	Moduł przeszedł pomyślnie konfiguracje3, odebrano konfiguracje Config3
3	Moduł przeszedł pomyślnie sprawdzenie sumy kontrolnej pamięci flash programu
4	Moduł przeszedł pomyślnie sprawdzenie sumy kontrolnej pamięci EEPROM
5	Uzyskany adres modułu jest identyczny z adresem zapamiętanym w pamięci EEPROM
6	Moduł przeszedł pomyślnie zerowanie czujnika siły
7	Moduł odczytał z pamięci EEPROM wartość zerową pozycji napędu
8	Brak błędu nasycenia/odłączenia czujnika siły
9	Brak błędu komunikacji z czujnikiem kata bezwzględne
10	Brak błędu magnesu czujnika kata bezwzględne
11	Brak błędu czujnika liniowego bezwzględne - jeżeli jego obecność ustawiona w konfiguracji węzła

12	Brak błędu odbicia światła dla bezwzględnego czujnika linowego
13	Wykryto podłączenie z modułem EMG
14	Wolne
15	Brak błędów czujników Halla silnika BLDC
16	Praca silnika (PWM > 0)
17	Wykrycie aktywacji linii alarmowej
18	Wykrycie błędu hamulca, hamulec jest zasilony, ale nie wykryto prądu hamulca
19	Przekroczenie ustalonej siły nacisku force_max
20	Aktywacja ograniczenia wypełnienia PWM silnika pwm_max
21	Aktywacja ograniczenia prądu silnika current_max_cont
22	Wykrycie napięcia zasilnia silnika i hamulca poniżej voltage_min
23	Wykrycie zadziałania ograniczenia pozycji kątowej dla position_min
24	Wykrycie zadziałania ograniczenia pozycji kątowej dla position_max
25	Wykrycie zadziałania ograniczenia pozycji liniowej dla actuator_min
26	Wykrycie zadziałania ograniczenia pozycji liniowej dla actuator_max
27	Wykrycie przekroczenia temperatury temp_driver_max
28	Wykrycie przekroczenia temperatury temp_motor_max
29	Aktywacja ogranicznika zakresu, synchronizowanego z innym węzłem
30	Wykryto odklejenie elektroniki EMG kanał 1
31	Wykryto odklejenie elektroniki EMG kanał 2

Tab. 4.20 prezentuje zestaw parametrów cyklicznie zwracanych z każdego modułu Slave do aplikacji sterującej pracą systemu robota rehabilitacyjnego.

Tab. 4.20 Opis pakietu ControlReplay (0xB1)

Bajt	Nazwa parametru (numer bajtu)	Typ	Opis	Uwagi
0.	mode (0)	uint32	Tryb pracy węzła: bity 3-0, opis w Tab. 4.16	
1.	mode (1)		Tryb pomiaru siły: bity 6-4, opis w Tab. 4.17	
2.	mode (2)		Strona pracy: bity 8-9, opis w Tab. 4.18	
3.	mode (3)			
4.	warnings (0)	uint32	Kody błędów, opis w Tab. 4.19	
5.	warnings (1)			
6.	warnings (2)			
7.	warnings (3)			
8.	force (0)	int16	Pomiar siły nacisku	Rozdzielczość: 0,1[N] Zakres: ±100,0[N]
9.	force (1)			
10.	emg_1 (0)	uint16	Pomiar amplitudy EMG kanał 1 (protagonista)	
11.	emg_1 (1)			
12.	emg_2 (0)	uint16		

13.	emg_2 (1)		Pomiar amplitudy EMG kanał 2 (antagonista)	
14.	position_a (0)	int16	Pomiar pozycji kątowej za pomocą enkodera bezwzględnego	Rozdzielczość: 0.1[°]
15.	position_a (1)			Zakres: ±180[°]
16.	position_h (0)	int32	Pomiar pozycji kątowej za pomocą czujników Hall'a	Rozdzielczość: 0.1[°]
17.	position_h (1)			Zakres: ±360.0[°]
18.	position_h (2)			
19.	position_h (3)			
20.	position_e (0)	int32	Pomiar pozycji kątowej za pomocą enkodera silnika (opcja)	Rozdzielczość: 0.1[°]
21.	position_e (1)			Zakres: ±360.0[°]
22.	position_e (2)			
23.	position_e (3)			
24.	speed_h (0)	int16	Pomiar prędkości obrotowej za pomocą czujników Halla	Rozdzielczość: 0.1[obr/min]
25.	speed_h (1)			Zakres: ±100.0[obr/min]
26.	speed_e (0)	int16	Pomiar prędkości obrotowej za pomocą enkodera silnika (opcja)	Rozdzielczość: 0.1[obr/min]
27.	speed_e (1)			Zakres: ±100.0[obr/min]
28.	temp_drv (0)	int16	Pomiar temperatury tranzystorów mocy	Rozdzielczość: 0,1[°C]
29.	temp_drv (1)			Zakres: 0 - 100,0[°C]
30.	temp_mot (0)	int16	Pomiar temperatury silnika (opcja)	Rozdzielczość: 0,1[°C]
31.	temp_mot (1)			Zakres: 0 - 100,0[°C]
32.	pwm (0)	int16	Odczyt wypełnienia PWM silnika	Rozdzielczość: 1
33.	pwm (1)			Zakres: ±1000
34.	current (0)	uint16	Pomiar prądu zasilania silnika	Rozdzielczość: 1[mA]
35.	current (1)			Zakres: 60000[mA]
36.	voltage (0)	uint16	Pomiar napięcia zasilania silnika	Rozdzielczość: 1[mV]
37.	voltage (1)			Zakres: 60000 [mV]
38.	actuator_a (0)	uint16	Pomiar rozsunięcia siłownika zmiany rozmiaru	Rozdzielczość: 1[mm]
39.	actuator_a (1)			Zakres: 0-100[mm]

Tab. 4.21 Opis pakietu Config1Replay (0xB2)

Bajt	Nazwa parametru (numer bajtu)	Typ	Opis	Uwagi
0	mode (0)	uint32	Tryb pracy wężła: bity 3-0, opis w Tab. 4.16	
1	mode (1)		Tryb pomiaru siły: bity 6-4, opis w Tab. 4.17	
2	mode (2)		Strona pracy: bity 8-9, opis w Tab. 4.18	
3	mode (3)			
4	position_min (0)	int32	Ustawienie ograniczenia pozycji (kąta) MIN	Rozdzielczość: 0.1[°]
5	position_min (1)			Zakres: ±180[°]
6	position_min (2)			
7	position_min (3)			

8	position_max (0)	int32	Ustawienie ograniczenia pozycji (kąta) MIN	Rozdzielczość: 0.1[°] Zakres: ±180[°]
9	position_max (1)			
10	position_max (2)			
11	position_max (3)			
12	speed_max (0)	uint16	Ustawienie ograniczenia maksymalnej prędkości	Rozdzielczość: 0.1[obr/min] Zakres: 100.0[obr/min]
13	speed_max (1)			
14	speed_rampa_up (0)	uint32	Ustawienie szybkości narastania prędkości	Rozdzielczość: 0.1[rpm/s] Zakres: 100[rpm/s]
15	speed_rampa_up (1)			
16	speed_rampa_up (2)			
17	speed_rampa_up (3)			
18	speed_rampa_down (0)	uint32	Ustawienie szybkości opadania prędkości	Rozdzielczość: 0.1[rpm/s] Zakres: 100[rpm/s]
19	speed_rampa_down (1)			
20	speed_rampa_down (2)			
21	speed_rampa_down (3)			
22	pwm_max (0)	uint16	Ustawienie ograniczenia PWM silnika	Rozdzielczość: 1 Zakres: 1000
23	pwm_max (1)			
24	current_max_pulse (0)	uint16	Ustawienie ograniczenia prądu chwilowego silnika (brak implementacji)	Rozdzielczość: 1 [mA] Zakres: 60000 [mA]
25	current_max_pulse (1)			
26	current_max_pulse_time (0)	uint16	Ustawienie czasu ograniczenia prądu chwilowego silnika (brak implementacji)	Rozdzielczość: 1 [ms] Zakres: 3000 [ms]
27	current_max_pulse_time (1)			
28	current_max_cont (0)	uint16	Ustawienie ograniczenia prądu ciągłego silnika	Rozdzielczość: 1[mA] Zakres: 60000 [mA]
29	current_max_cont (1)			
30	voltage_min (0)	uint16	Ustawienie minimalnego napięcia zasilania silnika	Rozdzielczość: 1[mV] Zakres: 60000 [mV]
31	voltage_min (1)			
32	force_gain (0)	uint32	Ustawienie wzmocnienia kanału pomiarowego siły	Rozdzielczość: 1 Zakres: 1000
33	force_gain (1)			
34	force_gain (2)			
35	force_gain (3)			
36	pwm_shift (0)	uint16	Ustawienie minimalnego wypełnienia PWM	Rozdzielczość: 1 Zakres: 1000
37	pwm_shift(1)			
38				
39				

Tab. 4.22 Opis pakietu Config2Replay (0xB3)

Bajt	Nazwa parametru (numer bajtu)	Typ	Opis	Uwagi
0	mode (0)	uint32	Tryb pracy węzła: bity 3-0, opis w Tab. 4.16	
1	mode (1)		Tryb pomiaru siły: bity 6-4, opis w Tab. 4.17	
2	mode (2)		Strona pracy: bity 8-9, opis w Tab. 4.18	
3	mode (3)			
4	pid_speed_p (0)	uint32		Rozdzielczość: 0,01

5	pid_speed_p (1)		Ustawienie regulatora PID – prędkość parametr P	Rozdzielczość: 0,01 Zakres: 10000
6	pid_speed_p (2)			
7	pid_speed_p (3)			
8	pid_speed_i (0)	uint32	Ustawienie regulatora PID – prędkość parametr I	Rozdzielczość: 0,01 Zakres: 10000
9	pid_speed_i (1)			
10	pid_speed_i (2)			
11	pid_speed_i (3)			
12	pid_speed_d (0)	uint32	Ustawienie regulatora PID – prędkość parametr D	Rozdzielczość: 0,01 Zakres: 10000
13	pid_speed_d (1)			
14	pid_speed_d (2)			
15	pid_speed_d (3)			
16	pid_speed_t (0)	uint32	Ustawienie regulatora PID – prędkość parametr T	Rozdzielczość: 1[ms] Zakres: 10000
17	pid_speed_t (1)			
18	pid_speed_t (2)			
19	pid_speed_t (3)			
20	pid_position_p (0)	uint32	Ustawienie regulatora PID – pozycja parametr P	Rozdzielczość: 0,01 Zakres: 10000
21	pid_position_p (1)			
22	pid_position_p (2)			
23	pid_position_p (3)			
24	pid_position_i (0)	uint32	Ustawienie regulatora PID – pozycja parametr I	Rozdzielczość: 0,01 Zakres: 10000
25	pid_position_i (1)			
26	pid_position_i (2)			
27	pid_position_i (3)			
28	pid_position_d (0)	uint32	Ustawienie regulatora PID – pozycja parametr D	Rozdzielczość: 0,01 Zakres: 10000
29	pid_position_d (1)			
30	pid_position_d (2)			
31	pid_position_d (3)			
32	pid_position_t (0)	uint32	Ustawienie regulatora PID – pozycja parametr T	Rozdzielczość: 1[ms] Zakres: 10000
33	pid_position_t (1)			
34	pid_position_t (2)			
35	pid_position_t (3)			
36				
...				
39				

Tab. 4.23 Opis pakietu Config3Replay (0xB4)

Bajt	Nazwa parametru (numer bajtu)	Typ	Opis	Uwagi
0	mode (0)	uint32	Tryb pracy węzła: bity 3-0, opis w Tab. 4.16	
1	mode (1)		Tryb pomiaru siły: bity 6-4, opis w Tab. 4.17	

2	mode (2)		Strona pracy: bity 8-9, opis w Tab. 4.18	
3	mode (3)			
4	temp_driver_max (0)	uint16	Ustawienie alarmu przekroczenia temperatury tranzystorów mocy	Rozdzielczość: 0,1[°C]
5	temp_driver_max (1)			Zakres: 0 - 100,0[°C]
6	temp_motor_max (2)	uint16	Ustawienie alarmu przekroczenia temperatury silnika (opcja)	Rozdzielczość: 0,1[°C]
7	temp_motor_max (3)			Zakres: 0 - 100,0[°C]
8	position_h_gear_ratio (0)	uint32	Ustawienie ilości impulsów z czujników Hall'a na pełny obrót wyjścia przekładni (przełożenie przekładni * ilość biegunów silnika * 6)	Rozdzielczość: 1
9	position_h_gear_ratio (1)			Zakres: 60000
10	position_h_gear_ratio (2)			
11	position_h_gear_ratio (3)			
12	position_e_gear_ratio (0)	uint32	Ustawienie ilości impulsów z enkodera silnika na pełny obrót wyjścia przekładni (przełożenie przekładni * ilość impulsów na obrót enkodera) (opcja)	Rozdzielczość: 1
13	position_e_gear_ratio (1)			Zakres: 60000
14	position_e_gear_ratio (2)			
15	position_e_gear_ratio (3)			
16	node_cfg_bits (0)	uint32	Ustawienie konfiguracji wężła	bit0 -negacja pozycji a kątowej
17	node_cfg_bits (1)			bit1 -negacja pozycji h kątowej
18	node_cfg_bits (2)			bit2 -negacja pozycji e kątowej
19	node_cfg_bits (3)			bit8 -podłączony hamulec bit9 -podłączony enkoder silnika bit10 -podłączony siłownik bit11 -podłączony enkoder liniowy bit12 -podłączony czujnik temperatury silnika
20				
...				
39				

Jak widać w

Tab. 4.24, moduł Slave umożliwia również odczyt informacji o numerach wersji oprogramowania wbudowanego modułu Ekspandera, modułu EMG, a także innych modułów, które mogą w przyszłości współpracować z modułem Slave robota. Takie rozwiązanie pozwala na pełną identyfikację elementów systemu sterowania przez oprogramowanie sterujące robotem i wyświetlenie tych informacji obsłudze lub serwisowi.

Tab. 4.24 Opis pakietu FirmwareVersionReplay (0xB5)

Bajt	Nazwa parametru	Typ	Opis	Uwagi
0	fv_main	5x char	Wersja firmware modułu Slave/Master numer wersji w postaci stringa: "0v00r" dla wersji Release lub "0v00d" dla wersji Debug	
1	fv_main			
2	fv_main			
3	fv_main			
4	fv_main			
5	fv_exp	5x char	Wersja firmware modułu Ekspandera numer wersji w postaci stringa: "0v00r" dla wersji Release lub "0v00d" dla wersji Debug	
6	fv_exp			
7	fv_exp			
8	fv_exp			
9	fv_exp			
10	fv_emg	5x char	Wersja firmware modułu EMG numer wersji w postaci stringa: "0v00r" dla wersji Release lub "0v00d" dla wersji Debug	
11	fv_emg			
12	fv_emg			
13	fv_emg			
14	fv_emg			
15	fv_empty1	5x char	Wersja firmware modułu XXXX numer wersji w postaci stringa: "0v00r" dla wersji Release lub "0v00d" dla wersji Debug	
16	fv_empty1			
17	fv_empty1			
18	fv_empty1			
19	fv_empty1			
20	fv_empty2	5x char	Wersja firmware modułu XXXX numer wersji w postaci stringa: "0v00r" dla wersji Release lub "0v00d" dla wersji Debug	
21	fv_empty2			
22	fv_empty2			
23	fv_empty2			
24	fv_empty2			
25	fv_empty3	5x char	Wersja firmware modułu XXXX numer wersji w postaci stringa: "0v00r" dla wersji Release lub "0v00d" dla wersji Debug	
26	fv_empty3			
27	fv_empty3			
28	fv_empty3			
29	fv_empty3			
30	fv_empty4	5x char	Wersja firmware modułu XXXX numer wersji w postaci stringa: "0v00r" dla wersji Release lub "0v00d" dla wersji Debug	
31	fv_empty4			
32	fv_empty4			
33	fv_empty4			
34	fv_empty4			
35	fv_empty5	5x char	Wersja firmware modułu XXXX numer wersji w postaci stringa: "0v00r" dla wersji Release	
36	fv_empty5			
37	fv_empty5			

38	fv_empty5		lub	
39	fv_empty5		"0v00d" dla wersji Debug	

Jak widać w Tab. 4.25, moduł Master odpowiada niewielką ilością danych w stosunku do modułu Slave. Zwracane dane pozwalają na identyfikację wersji oprogramowania modułu Master a także informują o stanie wyłącznika bezpieczeństwa, którego wciśnięcie prowadzi do zatrzymania pracy robota, której wznowienie jest możliwe dopiero po zwolnieniu wyłącznika bezpieczeństwa i potwierdzeniu intencji wznowienia pracy robota w aplikacji sterującej robotem. Dla celów testowych zwracany jest również pomiar czasu pomiędzy odbiorem kolejnych pakietów UDP przez moduł Master.

Tab. 4.25 Opis pakietu ControlMasterReplay (0xC1)

Bajt	Nazwa parametru (numer bajtu)	Typ	Opis	Uwagi
0	mode	uint8	Tryb pracy węzła MASTER, opis w Tab. 4.1	
1	status(0)	uint32	Flagi błędów i stanu pracy modułu Master bit0- 1-aktywny wyłącznik bezpieczeństwa	
2	status(1)			
3	status(2)			
4	status(3)			
5	period_time(0)	uint16	Czas pomiędzy poprzednim odpytaniem z PC	Rozdzielczość: 1[ms] Zakres: 65535
6	period_time(1)			
7	fv_main	5x char	Wersja firmware modułu Master numer wersji w postaci stringa: "0v00r" dla wersji Release lub "0v00d" dla wersji Debug	
8				
...				
39				

Podsumowując przedstawione tabelarycznie ramki danych i opisy pól bitowych, można opisać sekwencję uruchamiania i przejścia do wybranego trybu pracy robota, po którym aplikacja z komputera PC cyklicznie wysyła pakiety sterowania i oczekiwanie na pakiety odpowiedzi.

Sekwencja uruchamiania systemu:

1. Program na PC próbuje nawiązać połączenie z modułem Master, przez cykliczne wysyłanie pakietu **Control** (0xA1) co 1s, w tym czasie wszystkie moduły Slave w trybie **MODE_OFF**.
2. Po uzyskaniu odpowiedzi, w postaci pakietu **ControlReplay** (0xB1) od modułu Master, program PC przechodzi do sekwencji sprawdzająco-konfiguracyjnej.
3. Wysłane są zapytania o firmware **FirmwareVersion** (0xA5) do wszystkich węzłów, po którym następuje odbiór odpowiedzi **FirmwareVersionReplay** (0xB5).
4. Wysłane są pakiety konfiguracyjne **Config1** (0xA2) do wszystkich węzłów.
5. Wysłane są pakiety konfiguracyjne **Config2** (0xA3) do wszystkich węzłów.
6. Wysłane są pakiety konfiguracyjne **Config3** (0xA4) do wszystkich węzłów.
7. Sprawdzane są flagi błędów **NodeWarnings** wszystkich węzłach, moduły Slave pozostają w trybie: **MODE_OFF**.
8. Wysyłane są rozkazy zerowania czujników siły, za pomocą trybu: **MODE_CAL_FORCE**.
9. Wysyłane są rozkazy zerowania pozycji węzłów, za pomocą chwilowego przełączenia do trybu: **MODE_CAL_POSITION**. Co prowadzi do odczytania pozycji z bezwzględnych czujników pozycji i offsetów pozycji zapamiętanych w pamięci EEPROM podczas kalibracji. Na podstawie tych danych aktualizowany jest licznik pozycji względnej danego węzła.
10. Przejście do cyklicznego wysyłania pakietów **Control** (0xA1) do wszystkich węzłów systemu, po którym następuje odbiór komuników zwrotnych **ControlReplay** (0xB1). W tym stanie można realizować wybrane scenariusze pracy robota.

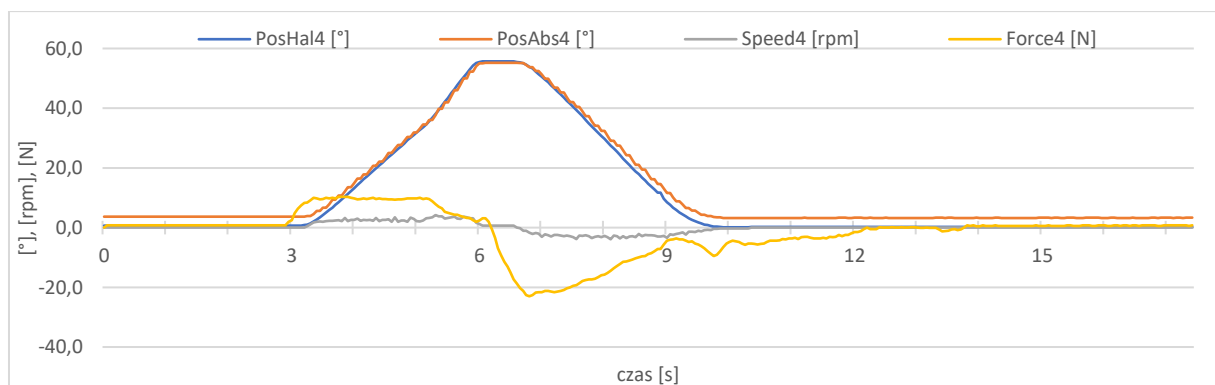
4.16. Rejestracja danych przez system sterowania robota

Istotną funkcjonalnością systemu sterowania robota rehabilitacyjnego jest możliwość rejestracji danych uzyskanych z elementów pomiarowych węzłów systemu. Podstawowym scenariuszem użycia tej funkcjonalności to rejestracja trajektorii ruchowych, w trybie śledzenia ruchu. A następnie, odtwarzanie zarejestrowanej trajektorii zadaną ilość razy, w celu osiągnięcia celu terapeutycznego.

Opracowana aplikacja komputerowa, sterująca pracą systemu sterowania robota rehabilitacyjnego, umożliwia zapisanie pliku z zarejestrowanymi parametrami podczas aktywnego trybu śledzenia ruchu, z domyślnym okresem próbkowania 50 ms. W skład zapisywanych parametrów wchodzi: licznik próbki czasowej wraz z parametrami zebranymi z węzłów systemu. W systemie sterowania robota rehabilitacyjnego jest to 7 węzłów. A każdy z węzłów reprezentowany jest przez 6 parametrów opisanych w pliku skrótami, które będą opisywały prezentowane parametry na kolejnych wykresach w niniejszej pracy, gdzie X oznacza numer węzła od 1 do 7:

- PosHalX – pozycja kątowa względna, wyliczana z czujników Halla silnika [°],
- PosAbsX – pozycja kątowa bezwzględna, odczytywana z enkodera magnetycznego [°],
- SpeedX – szybkość obrotu [rpm],
- ForceX – wartość odczytywana z czujnika siły [N],
- EmgAX – wartość napięcia odczytywana z kanały A modułu sEMG [mV],
- EmgBX – wartość napięcia odczytywana z kanały B modułu sEMG [mV].

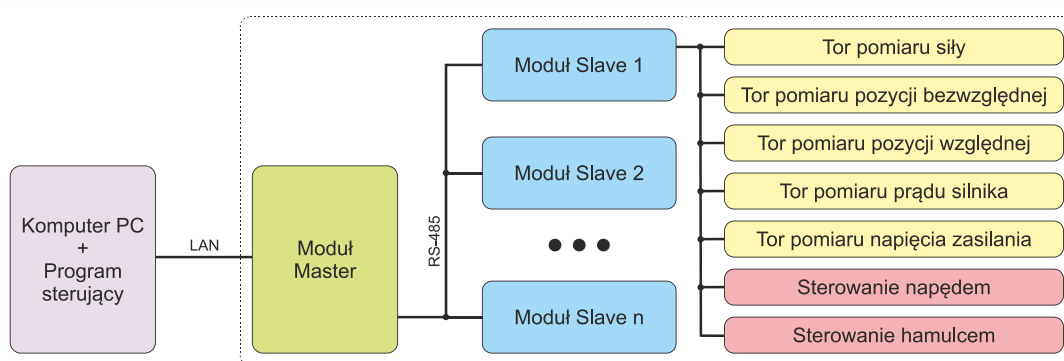
Zebrane dane mogą zostać wykorzystane do odtworzenia zapamiętanej trajektorii przez robota lub posłużyć do dalszej analizy zarejestrowanego ruchu, Rys. 4.51.



Rys. 4.51 Przykład wizualizacji danych z wybranego węzła systemu sterowania robota rehabilitacyjnego

5. WALIDACJA DZIAŁANIA SYTEMU STEROWANIA

System sterowania robota rehabilitacyjnego można podzielić na dwa komponenty. Pierwszym komponentem jest komputer z zainstalowaną aplikacją sterującą pracą robota. Drugim komponentem jest urządzenie w postaci robota rehabilitacyjnego, który połączony jest z komputerem sterującym za pomocą przewodu LAN z modulem Master robota. Elementami wykonawczymi robota są moduły Slave. Każdy moduł Slave steruje pracą napędu w pojedynczym węźle robota na podstawie zadanego trybu pracy z aplikacji sterującej PC i wartości pomiarowych uzyskanych z czujników współpracujących z danym węzłem, co schematycznie zostało zaprezentowane na Rys. 5.1.



Rys. 5.1 Uproszczony schemat blokowy systemu sterowania robota rehabilitacyjnego

Zapewnienie niezawodnej komunikacji i odpowiedniej dokładności torów pomiarowych jest istotne ze względów bezpieczeństwa pracy z robotem rehabilitacyjnym.

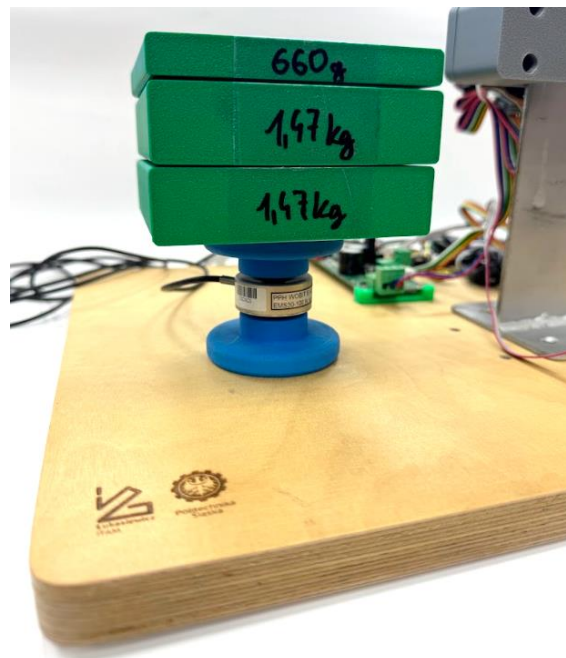
5.1. Walidacja układów pomiarowych systemu sterowania

5.1.1. Walidacja układu pomiarowego siły

Każdy moduł Slave jest wyposażony w tor pomiaru siły, jest to istotny element systemu sterowania, który mierzy siłę działającą pomiędzy pacjentem a robotem. W prezentowanej koncepcji systemu sterowania robota, siła jest mierzona w każdym uchwycie łączącym pacjenta z robotem. Czujnik siły, poza kontrolą wywieranej siły podczas odtwarzania ruchu, pozwala realizować tryb śledzenia ruchu. Dzięki czujnikowi siły jest możliwa realizacja ćwiczeń aktywnych, w których ruch jest możliwy po przekroczeniu określonej siły wywartej na dany

czujnik. Do pomiaru siły w robocie ARM-200 zostały zastosowane czujniki EMS30 firmy EMSYST w dwóch zakresach pomiarowych: $\pm 100\text{N}$ lub $\pm 200\text{N}$ [44], które charakteryzują się tolerancją czułości 2% i błędem nieliniowości 0,25%.

Walidacja toru pomiarowego współpracującego z czujnikiem siły została przeprowadzona za pomocą ciężarków o wagach: 0,044kg 0,660kg i 1,470kg. Obudowy ciężarków zostały wydrukowane na drukarce 3D i zasypane śrutem ołowianym, a następnie zostały zważone za pomocą wagi o dokładności $\pm 1\text{g}$. Dodatkowo, do czujnika siły o zakresie pomiarowym 100N zostały przykręcone wydrukowane krążki w kolorze niebieskim, widoczne na Rys. 5.2, umożliwiające ustawienie odważników na czujniku.



Rys. 5.2 Przedstawienie sposobu pomiaru siły nacisku

Badania zostały przeprowadzone po skalibrowaniu i wyzerowaniu toru pomiarowego siły. Wyniki pomiarów zostały przedstawione w Tab. 5.1. Jak widać, tor pomiarowy siły charakteryzuje się wysoką dokładnością pomiaru, błąd względny nie przekracza 0,2%, co jest bardzo dobrym wynikiem.

Tab. 5.1 Wyniki pomiaru dokładności toru pomiarowego siły

Lp.	Waga odważników	Nacisk odważników	Odczyt z systemu	Błąd bezwzględny	Błąd względny
	[kg]	[N]	[N]	[N]	[%]
1.	0,000	0,00	0,00	0,000	0,000
2.	0,044	0,43	0,36	-0,072	-0,166
3.	0,660	6,47	6,30	-0,175	-0,027
4.	1,470	14,42	13,98	-0,441	-0,031
5.	2,130	20,90	20,31	-0,585	-0,028
7.	2,940	28,84	28,62	-0,221	-0,008
8.	3,600	35,32	35,37	0,054	0,002

5.1.2. Walidacja układu pomiarowego prędkości kątowej

Pomiar prędkości w systemie sterowania robota jest realizowana przez pomiar czasu pomiędzy zmianami stanów na wyjściach czujnika Halla wbudowanych w silnik BLDC, zostało to przedstawione na Rys. 4.46. Im krótszy czas pomiędzy zmianami stanów tym większa prędkość. O prędkości wyjściowej napędu decyduje również przełożenie zastosowanej przekładni mechanicznej, a także ilość par biegunów silnika BLDC. Jest to pomiar cechujący się dużą powtarzalnością, ponieważ czas pomiędzy zmianami z czujników Halla jest mierzony przez mikrokontroler taktowany rezonatorem kwarcowym.

Referencyjne pomiary prędkości zostały wykonane za pomocą tachometru DT-2236, którego oś pomiarowa do realizacji pomiarów kontaktowych, została zaopatrzona w końcówkę stożkową. Następnie, na stanowisku badawczym, widocznym na Rys. 5.3, został uruchomiony pojedynczy moduł Slave współpracujący z napędem bez zamontowanego ramienia. Za pomocą aplikacji sterującej PC, zadawane były kolejne prędkości obrotowe do realizacji przez moduł Slave. Po ustawianiu nowej prędkości testowanego napędu, przykładany był tachometr do osi napędu i po ustabilizowaniu się wyniku na jego wyświetlaczu, odczytywany był pomiar prędkości kątowej.



Rys. 5.3 Sposób sprawdzenia dokładności pomiaru prędkości obrotowej

Wyniki badania dokładności pomiaru prędkości obrotowej zostały przedstawione w Tab. 5.2. Jak widać błąd względny nie przekracza 0,1%. Uzyskana dokładność pomiaru mieści się w przyjętych wymaganiach dla toru pomiarowego prędkości.

Tab. 5.2 Wyniki dokładności toru pomiaru prędkości obrotowej

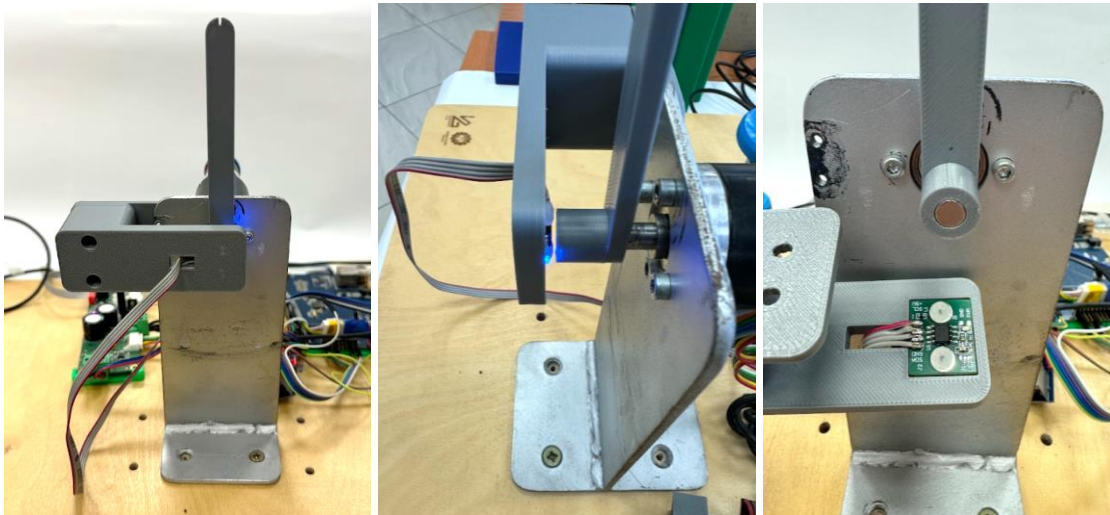
Lp.	Prędkość ustawiona	Odczyt z systemu	Odczyt z tachometru	Błąd bezwzględny	Błąd względny
	[rpm]	[rpm]	[rpm]	[rpm]	[%]
1.	1,0	1,0	0,9	-0,100	-0,100
2.	2,0	2,0	2,0	0,000	0,000
3.	3,0	3,0	3,0	0,000	0,000
4.	5,0	5,0	5,0	0,000	0,000
5.	10,0	10,0	10,0	0,000	0,000
6.	15,0	15,0	15,1	0,100	0,007
7.	20,0	20,0	20,1	0,100	0,005
8.	25,0	25,0	25,2	0,200	0,008

5.1.3. Walidacja układu pomiarowego kąta bezwzględnego

Czujnik kąta bezwzględnego, w systemie sterowania robotem rehabilitacyjnym, służy do ustalenia pozycji węzłów robota po włączeniu zasilania. Włączenie zasilania inicjuje odczytanie wartości pozycji kątowej z czujnika bezwzględnego, którą aktualizowany jest licznik względnej pozycji kątowej.

Można założyć, że pomiar kąta względnego jest praktycznie pomiarem cyfrowym wynikającego ze zliczania impulsów. Dlatego do oceny dokładności czujnika bezwzględnego został użyty licznik pozycji względnej. W analizowanym przypadku, licznik pozycji względnej charakteryzuje się rozdzielczością pozycji kątowej około $0,11^\circ$.

Badanie przeprowadzono na stanowisku pomiarowym, przedstawionym na Rys. 5.4, gdzie do osi przekładni przymocowano wydrukowane ramię z gniazdem na magnes trwały, współpracujący z badanym bezwzględnym czujnikiem kąta. Badanie przeprowadzono za pomocą systemu sterowania robota, który odczytywał równoległe pozycje kątowe z czujnika względnego i bezwzględnego. Pozycja kątowa była zmieniana przez ręczny obrót osi silnika.



Rys. 5.4 Zdjęcia magnetycznego czujnika bezwzględnego pozycji kątowej

Tab. 5.3 Wyniki pomiarów dokładności bezwzględnego czujnika pozycji kątowej

Lp.	Ustawienie czujnika względnego	Czujnik bezwzględny	Błąd bezwzględny	Błąd względny
	[°]	[°]	[°]	[%]
1.	100,0	101,3	1,300	0,013
2.	90,0	91,7	1,650	0,018
3.	80,0	81,9	1,850	0,023
4.	70,0	71,9	1,900	0,027
5.	60,0	62,1	2,100	0,035
6.	50,0	52,2	2,200	0,044
7.	40,0	42,0	2,000	0,050
8.	30,0	31,7	1,650	0,055
9.	20,0	21,2	1,150	0,057
10.	10,0	10,5	0,500	0,050
11.	0,0	-0,1	-0,050	
12.	-10,0	-10,7	-0,650	0,065
13.	-20,0	-21,2	-1,200	0,060
14.	-30,0	-31,5	-1,500	0,050
15.	-40,0	-41,7	-1,700	0,043
16.	-50,0	-51,7	-1,650	0,033
17.	-60,0	-61,5	-1,500	0,025
18.	-70,0	-71,2	-1,150	0,016
19.	-80,0	-80,8	-0,800	0,010
20.	-90,0	-90,4	-0,350	0,004
21.	-100,0	-99,9	0,150	-0,002

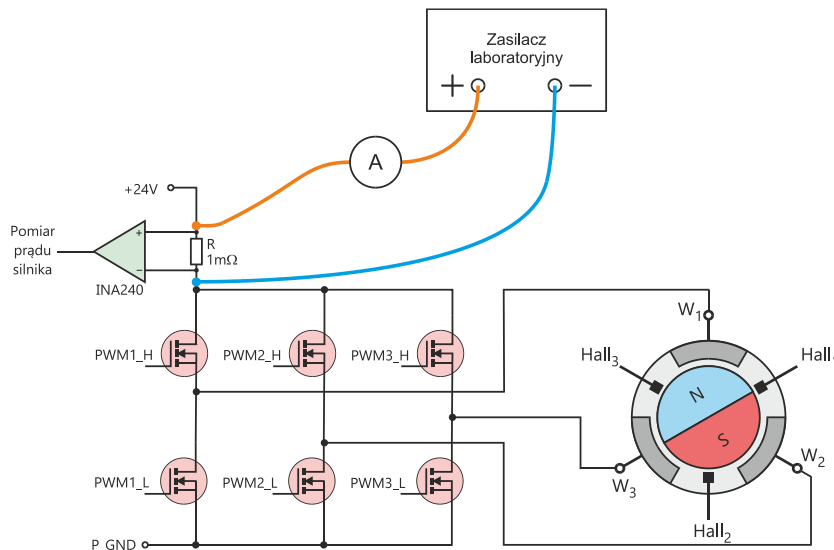
Jak widać w Tab. 5.3, maksymalny błąd pomiaru czujnika bezwzględnego nie przekracza wartości 2,2°, co mieści się w dopuszczalnym przydziale błędów $\pm 3^\circ$.

5.1.4. Badanie działania układu pomiaru prądu zasilania napędu

Tor pomiaru jest ważnym elementem systemu sterowania robota, jego rolą jest zabezpieczanie układu sterowania przed uszkodzeniem i co ważniejsze, pośredni pomiar siły wywieranej przez ramię robota na pacjenta lub na obiekty zewnętrzne wchodzące w interakcję z robotem.

Tor pomiarowy prądu silnika jest zrealizowany za pomocą rezystora pomiarowego o niskiej rezystancji $1\text{m}\Omega$, współpracującego ze specjalizowanym wzmacniaczem pomiarowym INA240, z którego sygnał analogowy podawany jest na wejście przetwornika

analogowo-cyfrowego. By zbadać dokładność toru pomiarowego prądu, do rezystora pomiarowego przyłączono zasilacz laboratoryjny w szereg z amperomierzem w postaci multimetru FLUKE 289, zgodnie z Rys. 5.5. Następnie zadawano kolejne wartości prądu za pomocą ograniczenia prądowego zasilacza laboratoryjnego, porównując wskazanie amperomierza referencyjnego z wartością prądu zmierzoną przez moduł Slave systemu sterowania robota. Pomiarzy zrealizowano w przedziale od 0 A do 5 A.



Rys. 5.5 Schemat badania dokładności toru pomiarowego prądu

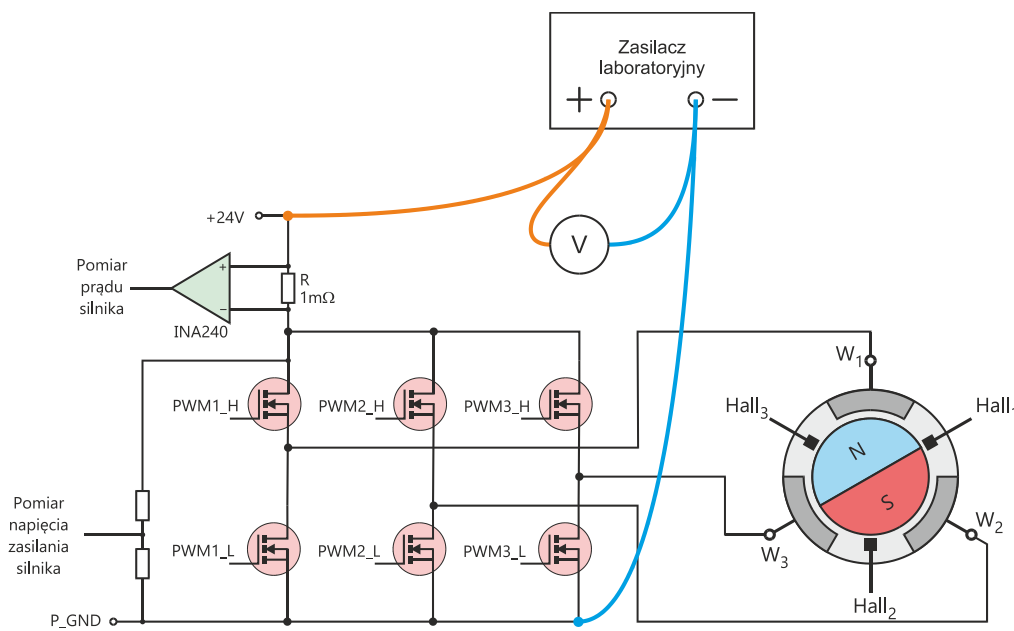
Wyniki pomiarów zostały przedstawione w Tab. 5.4, jak widać duży błąd prawie 12% występuje przy wartości prądu 1 A, co ważne z punktu widzenia bezpieczeństwa, ze wzrostem prądu błąd spada do wartości poniżej 1%. Duży błąd przy małych wartościach prądu prawdopodobnie wynika z zastosowania bardzo małej rezystancji rezystora pomiarowego, zwiększenie rezystancji powinno ten błąd zmniejszyć. Choć z punktu widzenia wykorzystania tego pomiaru, istotny jest mały błąd przy dużych wartościach prądu.

Tab. 5.4 Wyniki badania dokładności pomiaru prądu silnika BLDC

Lp.	Prąd zadany	Prąd zmierzony	Błąd bezwzględny	Błąd względny
	[A]	[A]	[A]	%
0	0,000	0,00	0,00	0,0
1	0,998	0,88	0,12	11,8
2	1,999	1,90	0,10	5,0
3	3,000	2,92	0,08	2,7
4	4,001	3,94	0,06	1,5
5	5,002	4,96	0,04	0,8

5.1.5. Badanie działania układu pomiaru napięcia zasilania napędu

Pomiar napięcia zasilania silnika BLDC w systemie sterowania robota rehabilitacyjnego ma na celu wykrycie sytuacji, w której napięcie zasilania znacznie spadnie poniżej ustawionego progu, co może być informacją o: uszkodzeniu przewodu lub gniazda zasilającego dany węzeł systemu, lub awarią zasilacza mocy. Praca w takim stanie może prowadzić do uszkodzenia innych elementów elektronicznych lub zagrożenia pożarowego. Dlatego ustawienie flagi wykrycia obniżonego napięcia zasilania blokuje dalszą pracę systemu.



Rys. 5.6 Schemat badania dokładności toru pomiarowego napięcia

Podczas badania porównywano napięcie zmierzone przez woltomierz referencyjny jakim był multimetr FLUKE 289, z napięciem zmierzonym przez tor pomiarowy napięcia zasilania silnika wbudowany w moduł Slave, zgodnie z Rys. 5.6.

Układ pomiaru napięcia jest skalibrowany w taki sposób by błąd był najmniejszy dla napięcia znamionowego silnika BLDC, jakim jest napięcie 24 V. Uzyskane wyniki zostały przedstawione w Tab. 5.5. Jak widać w zakresie od 0 V do 26 V błąd nie przekracza 0,2 V, co dla tego pomiaru jest wystarczającą dokładnością.

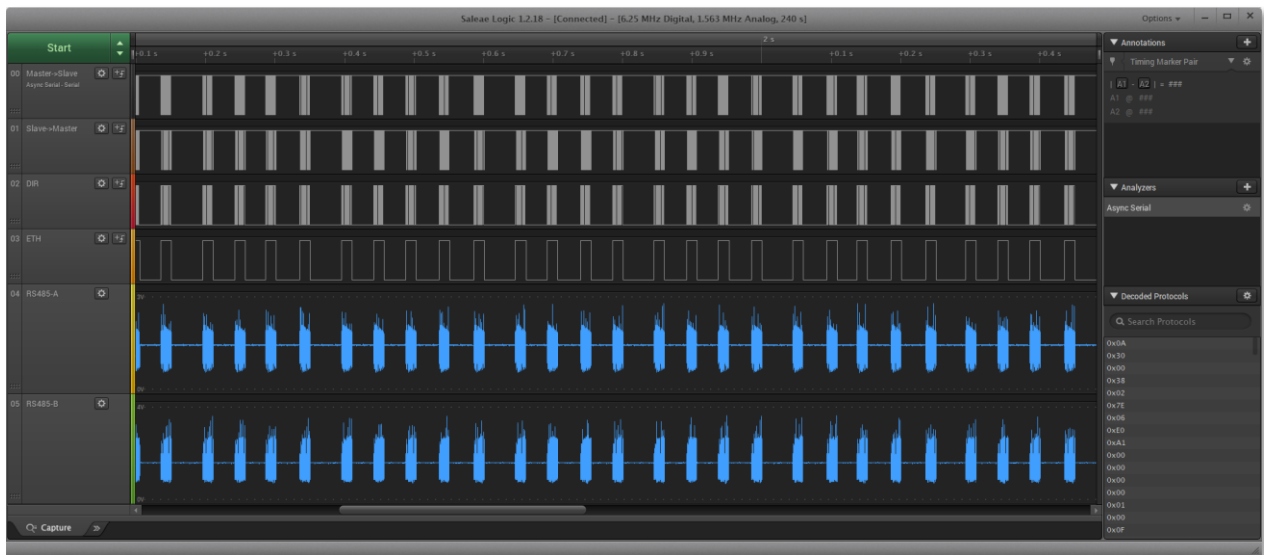
Tab. 5.5 Wyniki badania dokładności pomiaru napięcia zasilania

Lp.	Napięcie zadane	Napięcie zmierzone	Błąd bezwzględny	Błąd względny
	[V]	[V]	[V]	%
0	0,000	0,00	0,00	0,0
1	0,998	0,91	0,09	8,8
2	1,999	1,93	0,07	3,5
3	3,000	2,95	0,05	1,7
4	4,001	3,97	0,03	0,8
5	5,002	4,98	0,02	0,4
6	6,004	6,01	-0,01	-0,1
7	7,004	7,02	-0,02	-0,2
8	8,006	8,04	-0,03	-0,4
9	9,006	9,06	-0,05	-0,6
10	10,008	10,08	-0,07	-0,7
11	11,008	11,10	-0,09	-0,8
12	12,009	12,12	-0,11	-0,9
13	13,011	13,13	-0,12	-0,9
14	14,012	14,15	-0,14	-1,0
15	15,013	15,17	-0,16	-1,0
16	16,014	16,18	-0,17	-1,0
17	17,014	17,19	-0,18	-1,0
18	18,016	18,18	-0,16	-0,9
19	19,017	19,19	-0,17	-0,9
20	20,018	20,18	-0,16	-0,8
21	21,019	21,16	-0,14	-0,7
22	22,021	22,14	-0,12	-0,5
23	23,021	23,09	-0,07	-0,3
24	24,023	24,03	-0,01	0,0
25	25,023	24,95	0,07	0,3
26	26,024	25,84	0,18	0,7
27	27,025	26,70	0,32	1,2
28	28,026	27,54	0,49	1,7
29	29,020	28,34	0,68	2,3
30	30,021	29,11	0,91	3,0

5.2. Analiza wpływu zastosowania systemu operacyjnego nie czasu rzeczywistego (ang.: non-real-time operating system)

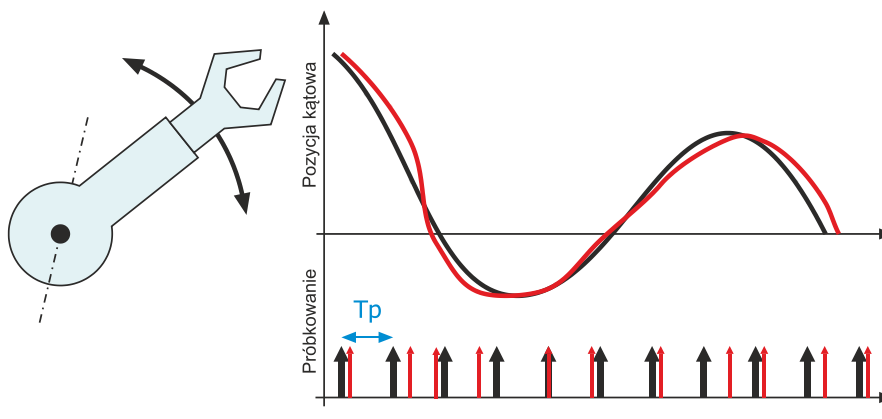
Ze względu na zakładaną współpracę systemu sterowania robota z oprogramowaniem do realizacji m.in. do gier rehabilitacyjnych, które wykorzystuje środowisko .NET w systemie operacyjnym Microsoft Windows 10, przeprowadzono badania mające na celu ocenę potencjalnego negatywnego wpływu takiego rozwiązania na pracę robota. Podczas badania mierzono zmienność taktowania kolejnych przesłań danych pomiędzy aplikacją PC uruchomioną w systemie operacyjnym Microsoft Windows 10, a systemem sterowania robota rehabilitacyjnego ARM-200. Ocenę wpływu przeprowadzono na stanowisku badawczym zbudowanym z systemu sterującego pojedynczym stopniem swobody robota.

Wstępne badanie przeprowadzono za pomocą analizatora stanów logicznych Saleae Logic Pro 8, który został podłączony do sygnałów magistrali RS-485. Stan wysoki sygnału ETH informuje o przyjęciu pakietu UDP z oprogramowania sterującego robotem i obsłudze cyklu wymiany danych z modułami robota, który zakończony jest wysłaniem pakietu zwrotnego UDP i zmianą stanu linii ETH na niski. Stan wysoki sygnału ETH pozwala określić rzeczywisty czas pełnej obsługi cyklu wymiany danych w systemie sterowania robotem. A czasu obecności stanu niskiego sygnału ETH pozwala ocenić zmienność okresu taktowania komunikacji inicjowanej przez oprogramowanie PC. Na Rys. 5.7 został przedstawiony wynik przechwycenia komunikacji za pomocą analizatora stanów logicznych, na którym wyraźnie widać, że czas powtarzania komunikacji T_p podlega widocznej zmienności. Uzyskanie takiego obrazu zmienności czasu T_p było pretekstem do postawienia pytania: czy takie wahania czasu T_p mogą wpływać w istotny sposób na dokładność rejestracji i odtwarzania ruchu przez robota rehabilitacyjnego. Problem braku precyzji i powtarzalności taktowania procesów w systemach operacyjnych czasu nie rzeczywistego jest zjawiskiem oczywistym, jednak w wielu przypadkach nie wpływa znacząco na działanie aplikacji. Podejmowane są również próby odpowiedzi na pytanie, w jakich językach [63] i z użyciem jakich mechanizmów odmierzenia czasu [64] można uzyskać najdokładniejszy pomiar czasu.



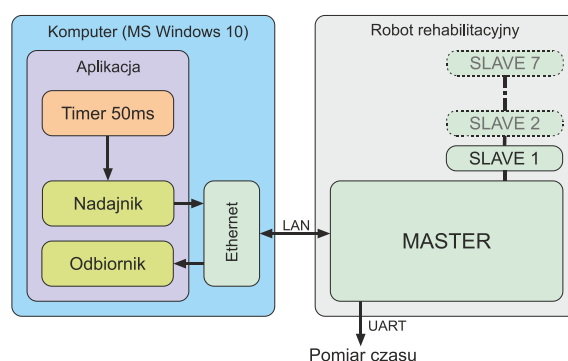
Rys. 5.7 Widok zmienności czasu powtarzania T_p

W prezentowanym przypadku można założyć, że jeżeli ruch ramienia robota w pojedynczym stopniu swobody zostanie zadany np. syntetycznie ruchem spróbkowanym w równych odstępach czasu, a następnie chwilowe pozycje kątowe ramienia zostaną przesłane do węzła wykonawczego w nieregularnych odstępach czasu to odtwarzany ruch zostanie zniekształcony Rys. 5.8.



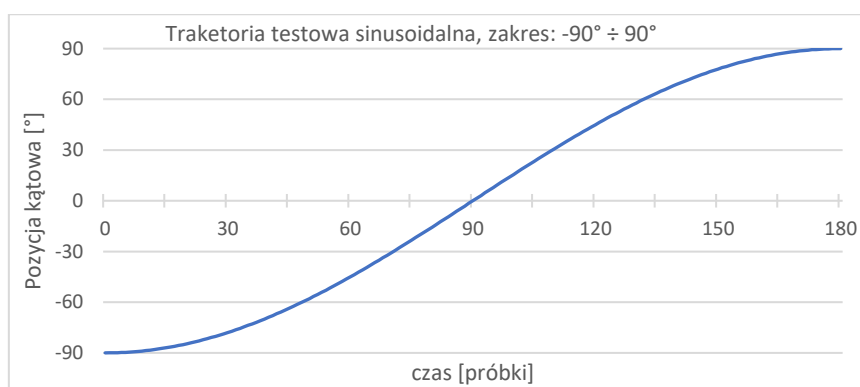
Rys. 5.8 Rysunek prezentuje hipotetyczny efekt nieregularnej komunikacji

Pomiar czasu powtarzania T_p został zrealizowany za pomocą mikrokontrolera (ARM Cortex M4 taktowany zegarem 120MHz [59]) modułu Master, gdzie wewnętrzny timer zliczał czas pomiędzy kolejnymi odbiorami danych z aplikacji sterującej na PC. Czas był zliczany z rozdzielczością 1 ms. By umożliwić odczyt pomiaru czasu, moduł Master został wyposażony w dodatkowy kanał komunikacji w postaci portu szeregowego, na który wysyłane były wartości zmierzonego czasu, pomiędzy odbieranymi pakietami z interfejsu LAN (Rys. 5.9).



Rys. 5.9 Schemat przepływu informacji podczas eksperymentu

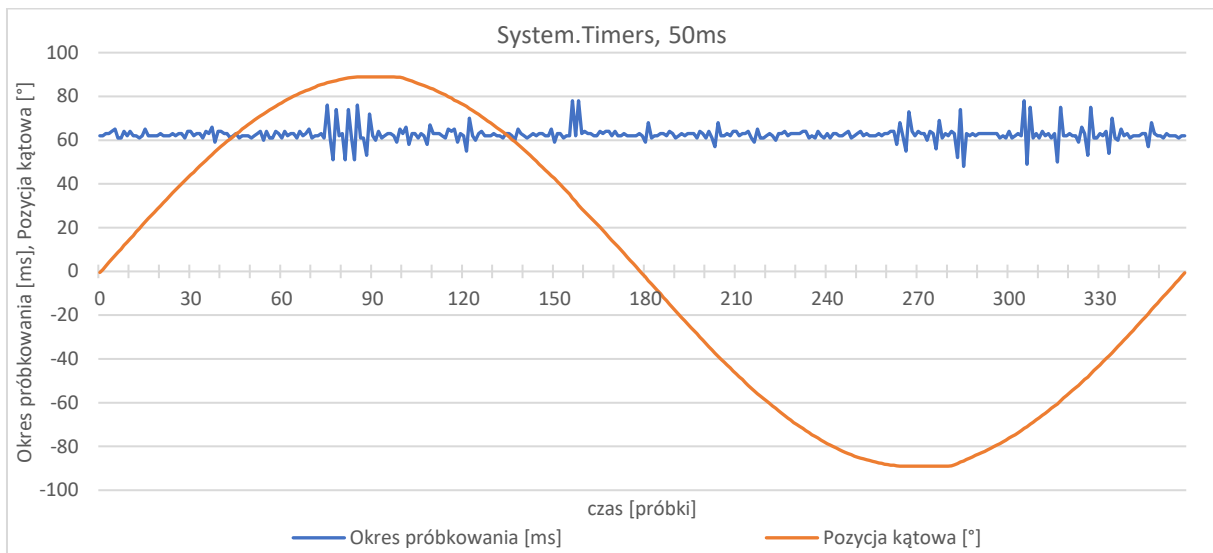
Do eksperymentu został przygotowany syntetyczny trening (Rys. 5.10) w postaci sinusoidalnej trajektorii ruchu, w którym połowa okresu została podzielona na 180 kroków. Aplikacja sterująca umożliwia zapętlenie ruchu, tak by po osiągnięciu ostatniej próbki nastąpiło odwrócenie kierunku odtwarzania trajektorii ruchowej. Okres pomiędzy wysłaniem kolejnego kroku to czas T_p , w tym wypadku ustalony na 50ms.



Rys. 5.10 Syntetyczna trajektoria ruchu ramienia robota (zapętłona w aplikacji sterującej)

Program sterujący pracą robota została napisana w języku C# w środowisku Microsoft Visual Studio jako aplikacja typu Windows Form (.NET Framework). Badanie przeprowadzono dla dwóch metod pomiaru czasu. W pierwszej metodzie zastosowano standardową klasę Timer (System.Timers). Drugą metodą było zastosowanie klasy Stopwatch (System.Diagnostics.Stopwatch). Analizę przeprowadzono dla ruchu w pełnym okresie funkcji sinus, czyli dla 360 próbek treningu. Przy założeniu próbkowania z okresem 50ms pełny trening powinien trwać 18 sekund.

Rys. 5.11 przedstawia zestawienie czasu pomiędzy próbkami wysłanej trajektorii z pozycją kątową ramienia robota, odczytaną z enkodera wbudowanego w silnik ramienia robota. Jak widać w wielu miejscach wartość opóźnienia przekracza dopuszczalną wartość opóźnienia wynoszącą 50 ms. Wyznaczona z zarejestrowanego przebiegu średnia wartość wynosi 62,598 ms.

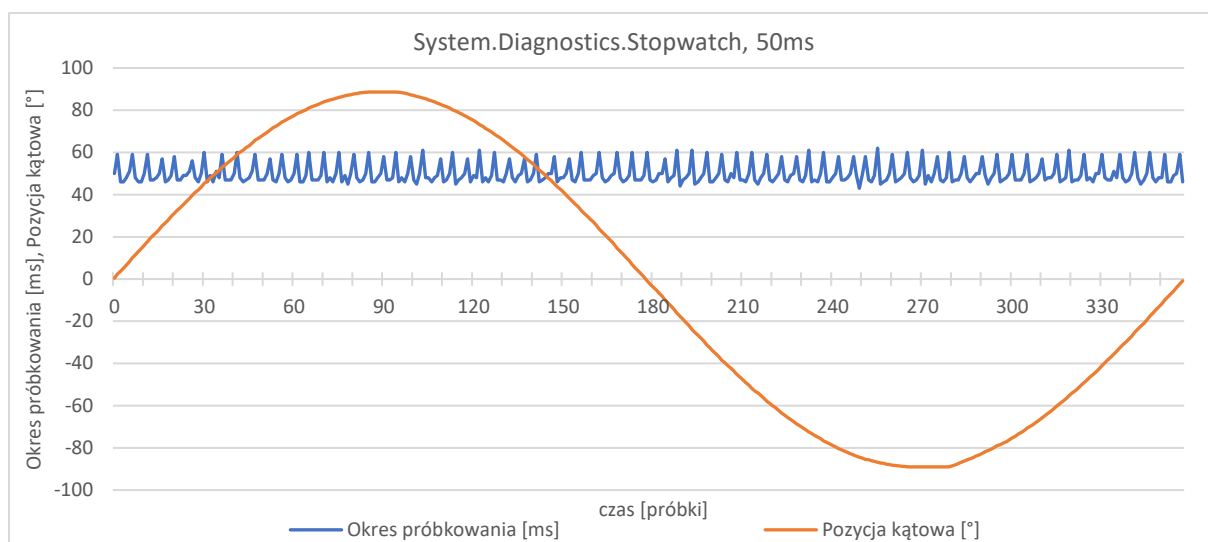


Rys. 5.11 Wykresy czasu pomiędzy próbkami i pozycji kątowej ramienia robota dla klasy Timers

Efektom wydłużenia wartości średniej okresu próbkowania jest wydłużenie odtwarzanego treningu. Jeżeli przykładowy trening trwał 10 min to jego czas, przy wyznaczonej średniej próbkowania 62,598 ms, zostanie wydłużony do 12,5 min. Co może mieć już istotne znaczenie terapeutyczne.

Natomiast na podstawie obserwacji wykresu zmian pozycji kątowej, nie można zauważyć wpływu zmienności czasu próbkowania na trajektorię ruchu ramienia robota.

Wyniki uzyskane przy pomocy drugiej metody taktowania czasu wysyłania komunikatów z aplikacji sterującej PC przedstawia Rys. 5.12. W tym przypadku, również widać zmienność czasu próbkowania, jednak jest ona bardziej regularna w porównaniu do pierwszej metody. Wartość średnia czasu próbkowania wynosi 50,014 ms co jest czasem praktycznie idealnym, nie wpływającym na czas trwania odtwarzanego treningu.



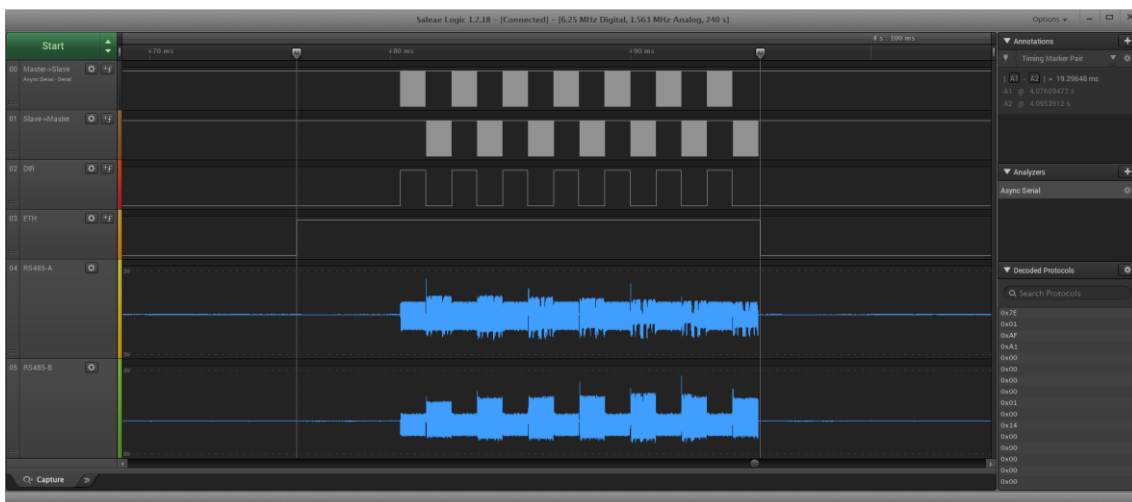
Rys. 5.12 Wykresy czasu pomiędzy próbkami i pozycji kątowej ramienia robota dla klasy Stopwatch

W obu analizowanych przypadkach, zmierzona zmienność próbkowania trajektorii ruchu w systemie sterowania robota rehabilitacyjnego nie wpływa w zauważalny sposób na zniekształcenie odtwarzanego ruchu. Odporność na chwilowe wahania częstotliwości próbkowania wynika z bezwładności regulatora, który skutecznie filtruje takie zakłócenia. W przypadku wykorzystania klasy Timers, zauważono negatywne zjawisko w postaci wydłużenia średniej wartości okresu próbkowania, co wpływa na wydłużenie odtwarzanych treningów w stosunku do treningów zaprojektowanych. Natomiast, jeżeli rejestracja ruchu i późniejsze odtwarzania ruchu będzie taktowane tą samą metodą to nie wpłynie to negatywnie na zmianę czasu odtwarzania treningu.

Niniejsza praca nie analizuje wszystkich negatywnych aspektów mogących wystąpić podczas sterowania robota rehabilitacyjnego za pomocą aplikacji uruchomionej w systemie operacyjny Windows. W dalszej analizie tego problemu, można również zbadać wpływ zmienności taktowania komunikacji w zależności od mocy obliczeniowej zastosowanego komputera, a także zbadać wpływ dodatkowego obciążenia komputera innymi aplikacjami uruchomionymi w systemie. By ograniczyć negatywny wpływ „wewnętrzznego życia” systemu operacyjnego Windows, warto zadbać o to, by na komputerze sterującym robotem rehabilitacyjnym nie były uruchomione inne aplikacje. Innym zaleceniem może być odłączenie komputera od sieci Internet, co zwiększy cyberbezpieczeństwo systemu, a także zapobiegnie niespodziewanej aktualizacji systemu, która wyłączy system z użytkowania w nieoczekiwanym momencie. Innym podejściem, zapewniającym stabilność taktowania próbkowania, może być zmiana filozofii sterowania, tak by to moduł Master taktował komunikację z komputerem.

5.3. Badanie szybkości komunikacji aplikacji sterującej z robotem

Głównym ograniczeniem częstości wymiany danych pomiędzy aplikacją sterującą a robotem jest szybkość magistrali szeregowej RS-485 i ilość danych wymienia w każdym cyklu komunikacji. W naszym przypadku prędkość magistrali RS-485 jest ustawiona na 460 800 bps i w każdym cyklu Master wysyła i odbiera po 48 bajtów. Co przy 7 węzłach daje w sumie 672 bajty. W idealnym przypadku, pojedynczy cykl wymiany danych wymaga czasu około 14,6 ms. Niestety, z uwagi na opóźnienia związane z przetwarzaniem danych i przełączaniem kierunku magistrali RS-485 realny czas wymiany danych jest dłuższy.



Rys. 5.13 Pomiar czasu wymaganego na obsługę pełnego cyklu wymiany danych

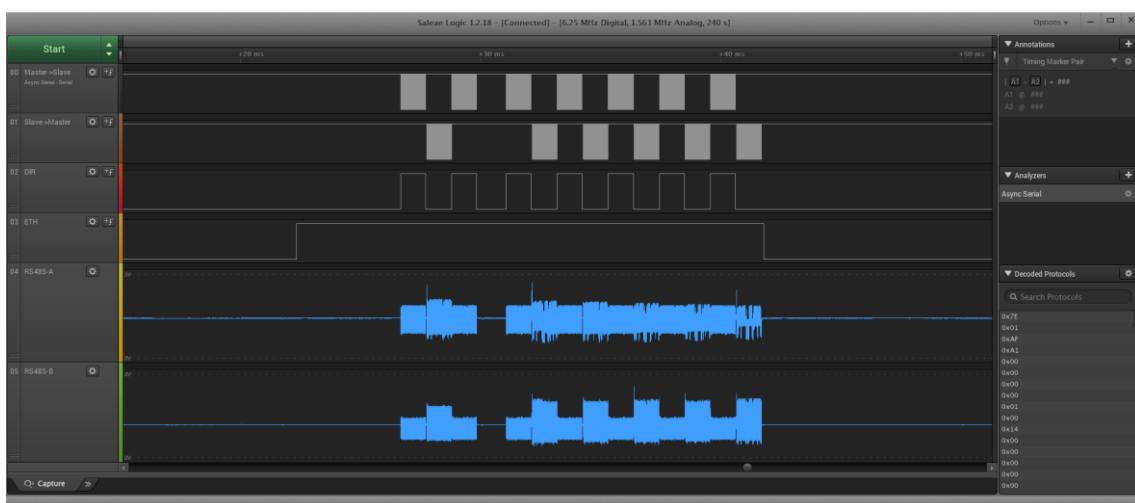
W badaniu tym, mierzono czas wymagany do realizacji pełnego cyklu wymiany danych pomiędzy aplikacją sterującą PC a węzłami wykonawczymi robota (moduły Slave). Pomiar czasu zrealizowano za pomocą analizatora stanów logicznych Saleae Logic Pro 8. Przykładowe wyniki zaprezentowano na Rys. 5.13, gdzie zwizualizowana została graficznie aktywność na liniach magistrali danych łączącej moduł Master z modułami Slave robota. Analizatora stanów został podłączony do sygnałów nadawania i odbioru interfejsu szeregowego modułu Master, a także do sygnału DIR, który steruje kierunkiem magistrali RS-485. Stan wysoki sygnału ETH informuje o przyjęciu pakietu UDP z oprogramowania sterującego robotem i obsłudze cyklu wymiany danych z modułami robota, który zakończony jest wysłaniem pakietu zwrotnego UDP i zmianą stanu linii ETH na niski. Stan wysoki sygnału ETH pozwala określić rzeczywisty czas pełnej obsługi cyklu wymiany danych w systemie sterowania robotem. Markery (A1 i A2) ustawione na zboczach sygnału ETH ułatwiają pomiar

czasu pełnego cyklu wymiany danych, który wynosi około 19,3ms. Uzyskany wynik z dużym zapasem spełnia założony czas na cykl wymiany danych wynoszący maksymalnie 100ms.

5.4. Badanie zachowania systemu sterowania w warunkach braku komunikacji z pojedynczym modułem Slave

Praca robota rehabilitacyjnego jest możliwa tylko w przypadku działania wszystkich jego węzłów. Ważne jest, by komunikacja z każdym węzłem była niezależna i odporna na zakłócenia chwilowe lub permanentne w pracy innych węzłów. By sprawdzić odporność opracowanego rozwiązania na uszkodzenie jednego z węzłów systemu, przeprowadzono eksperyment, w którym w trybie debug zatrzymano wykonywanie programu mikrokontrolera sterującego węzłem. Zatrzymanie programu wykonano po uruchomieniu wszystkich węzłów i przeprowadzeniu procedury adresowania węzłów. W innym przypadku, zgłoszony byłby błąd podczas uruchomienia systemu, z uwagi na sygnalizację zmiany adresu modułu. Ponieważ przy każdym uruchomieniu moduł sprawdza czy poprzednio moduł miał ten sam adres. Adres jest zapamiętywany w pamięci nieulotnej (EEPROM) każdego modułu Slave.

Analizę zachowania komunikacji przeprowadzono za pomocą analizatora stanów logicznych Saleae Logic Pro 8, który został podłączony do sygnałów magistrali RS-485. Jak można zauważyć na Rys. 5.14, na wykresie górnym, moduł Master wysyła rozkazy do 7 modułów Slave. Natomiast, wykres poniżej przedstawia transmisję danych zwrotnych z modułów Slave, gdzie brakuje odpowiedzi z modułu Slave nr 2.



Rys. 5.14 Reakcja systemu sterowania na brak odpowiedzi modułu Slave nr2

Jak widać na Rys. 5.14, moduł Master w przypadku komunikacji z modułem Slave nr 2 oczekuje na odpowiedź, a po jej braku przechodzi do dalszej komunikacji z kolejnymi modułami Slave. Wynik obserwacji potwierdza oczekiwaną odporność komunikacji na zakłócenia spowodowane utratą komunikacji z modułem Slave, ponieważ brak odpowiedzi jednego z modułów Slave nie przerywa cyklu wymiany danych z kolejnymi modułami.

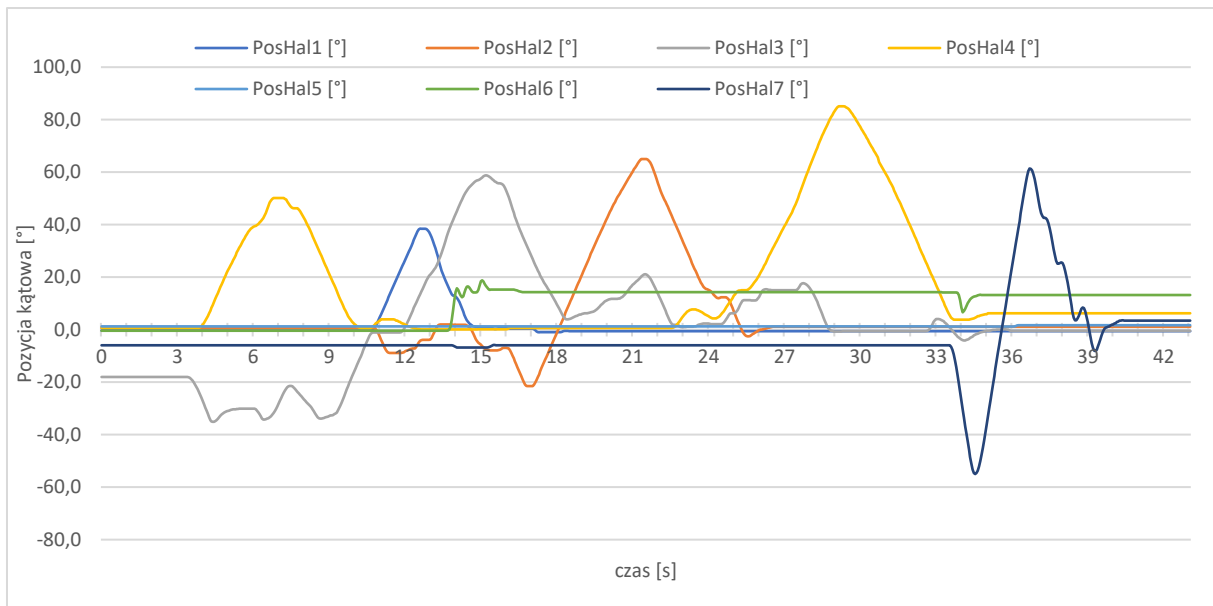
5.5. Testowanie działania robota w pełnej konfiguracji

W tym badaniu testowano działanie zaprojektowanego układu sterowania w trybie śledzenia ruchu, wywierając nacisk na kolejne czujniki w węzłach robota i uruchamiając odpowiadające im napędy. Scenariusz badania przeprowadzono w taki sposób, aby aktywować ruch w węzłach robota w kolejności odpowiadającej następującym ruchom kończyny górnej: zgięcie w stawie łokciowym, odwiedzenie kończyny, zgięcie w stawie nadgarstkowym, rotacja w stawie barkowym, zgięcie w stawie barkowym, powtórzone zgięcie w stawie łokciowym i rotacja w stawie łokciowym.



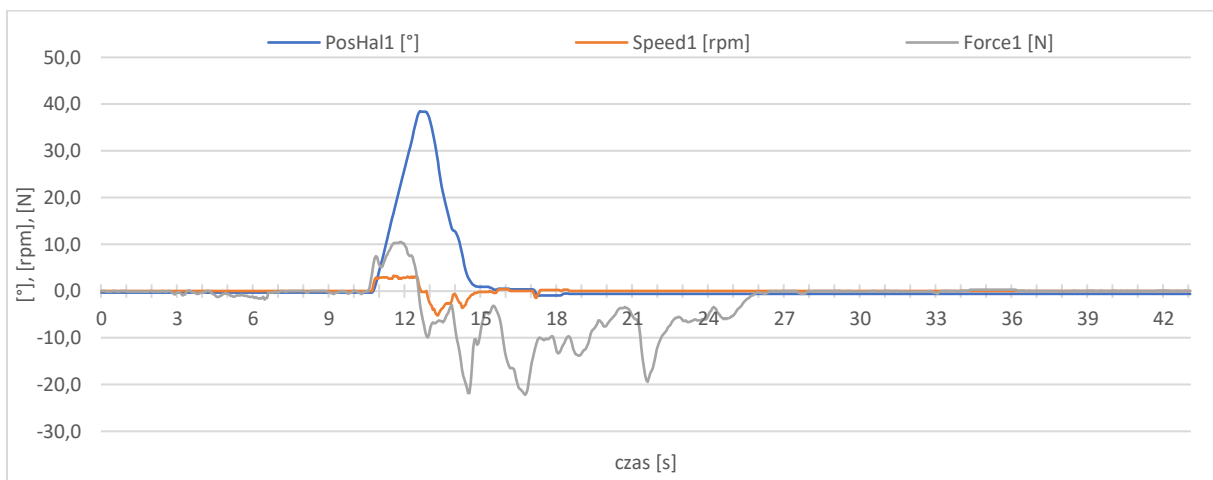
Rys. 5.15 Zdjęcie robota ARM-200 w konfiguracji na kończynę prawą

Przedstawione poniżej wykresy przedstawiają reakcję węzłów robota rehabilitacyjnego ARM-200 (Rys. 5.15) realizującego tryb śledzenia ruchu kończyny, dla numeracji węzłów wg Rys. 4.2. Podczas eksperymentu starano się wywołać ruch, po kolei, wszystkich węzłów robota z podobną prędkością, co prezentują wykresy na Rys. 5.16.

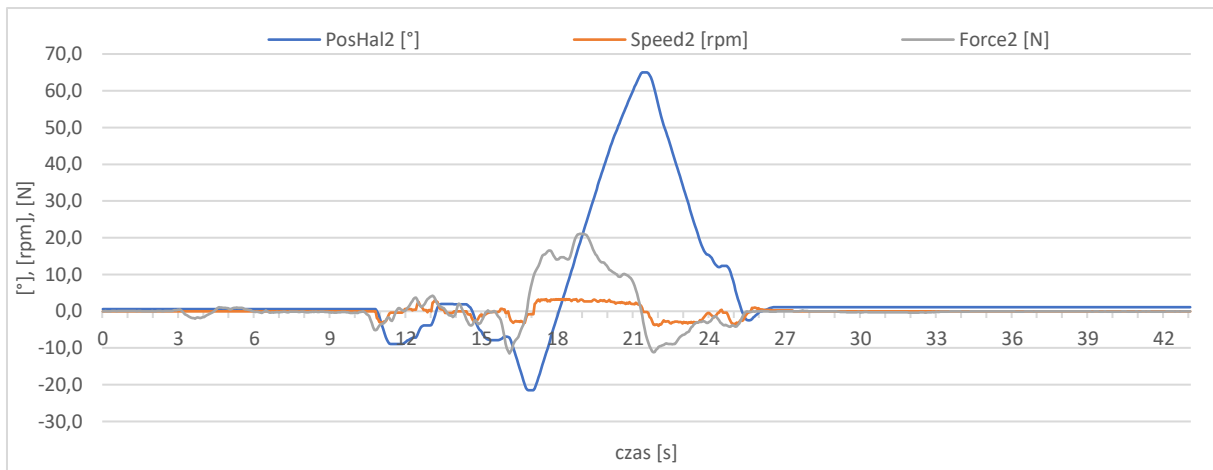


Rys. 5.16 Zmiany pozycji w 7 węzłach robota rehabilitacyjnego ARM-200

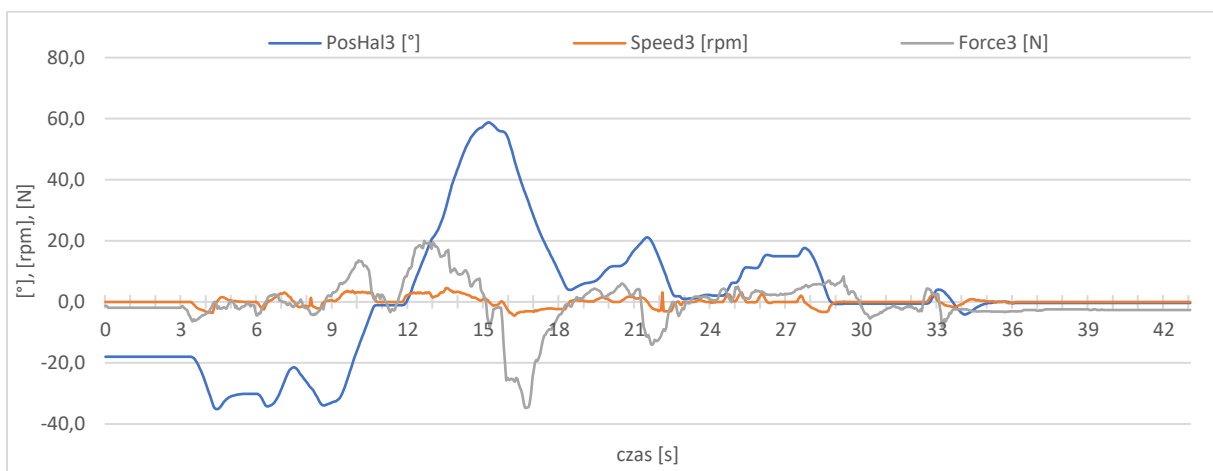
Na wykresach poniżej, na rysunkach od Rys. 5.17 do Rys. 5.23, przedstawiono zmiany parametrów węzłów (od 1 do 7) robota: pozycji względnej, prędkości ruchu i siły nacisku, podczas wykonywania ruchu testowego opisanego wyżej.



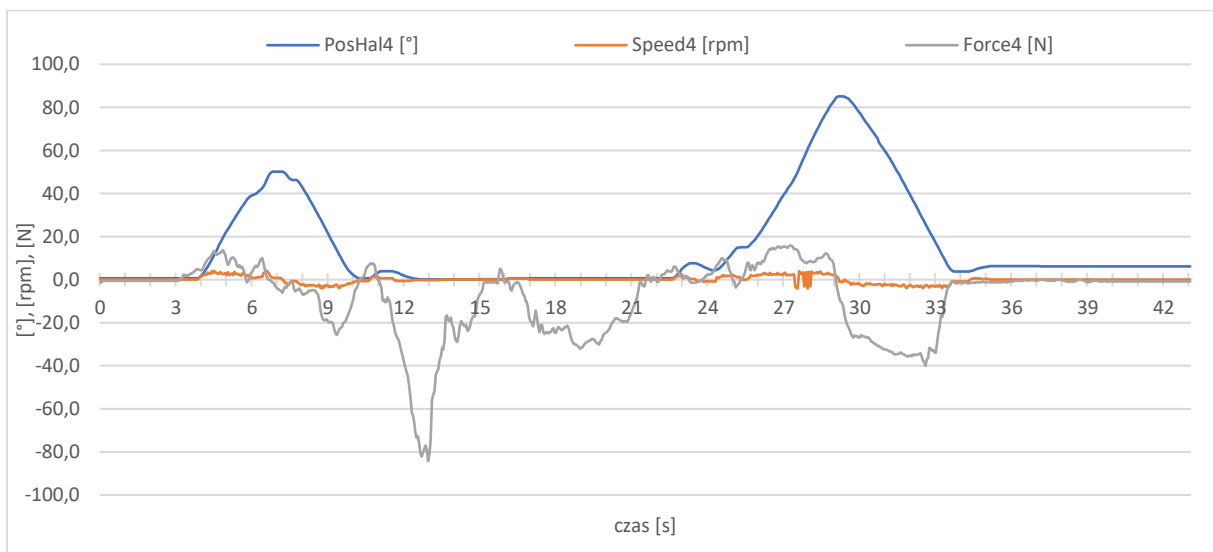
Rys. 5.17 Węzeł 1 - odwiedzenie w stawie barkowym



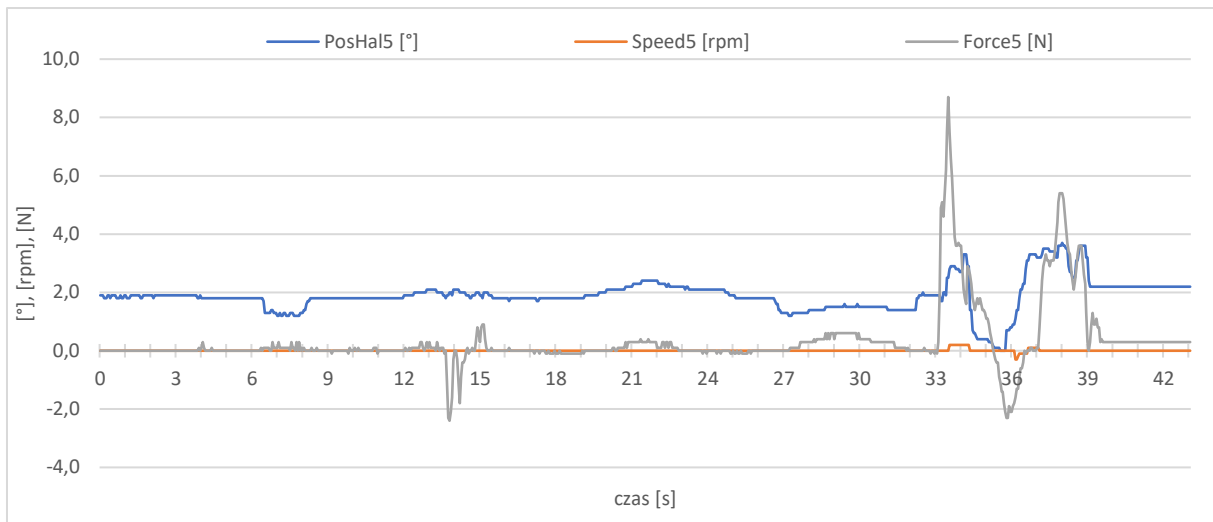
Rys. 5.18 Węzeł 2 - zgięcie w stawie barkowym



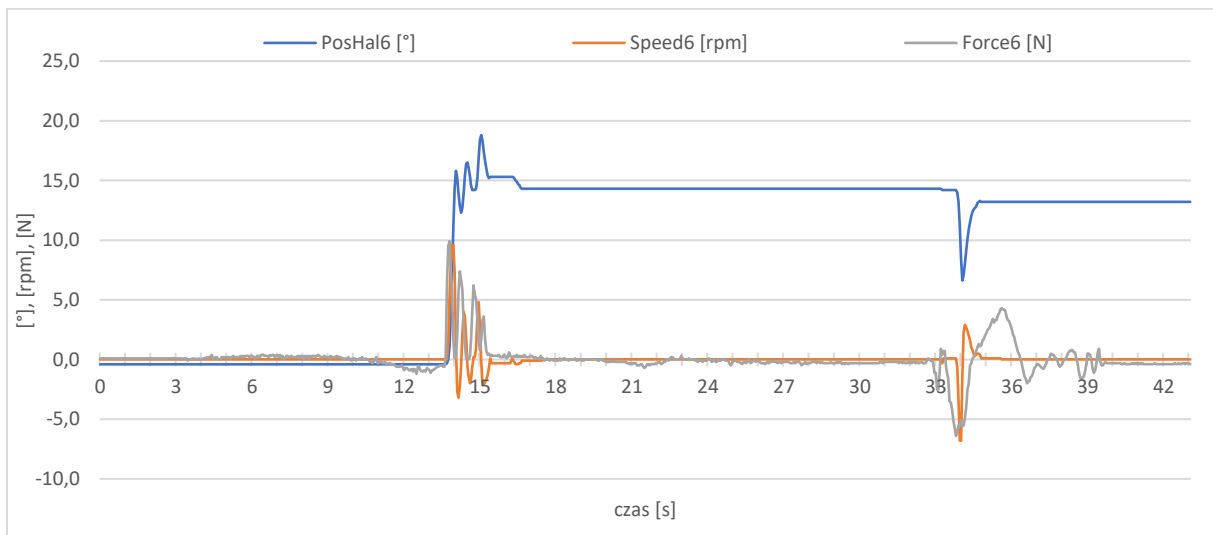
Rys. 5.19 Węzeł 3 - rotacja w stawie barkowym



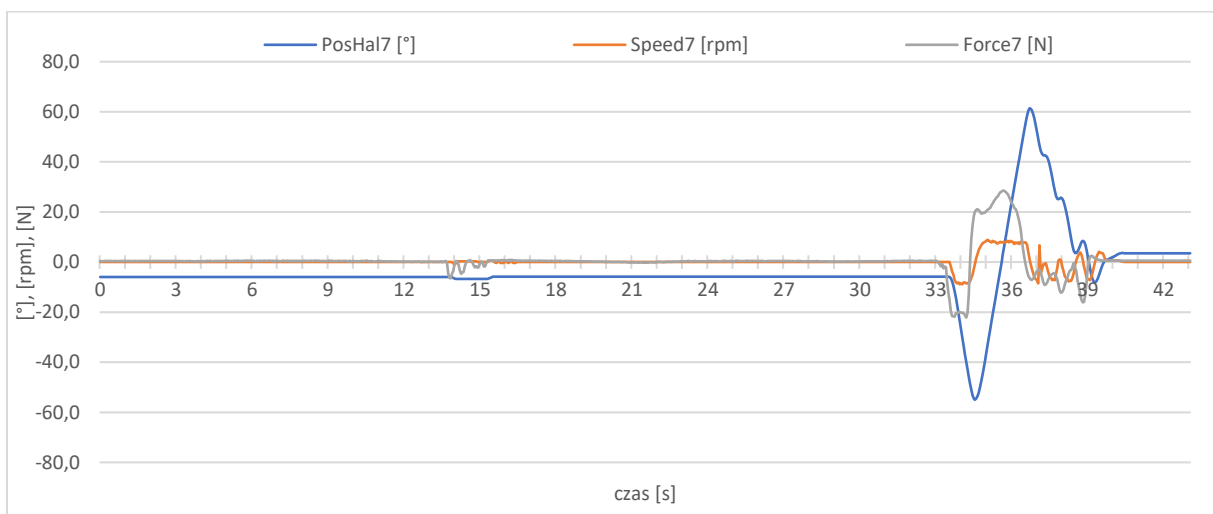
Rys. 5.20 Węzeł 4 - zgięcie w stawie łokciowym



Rys. 5.21 Węzeł 5 - zgięcie promieniowo-łokciowe



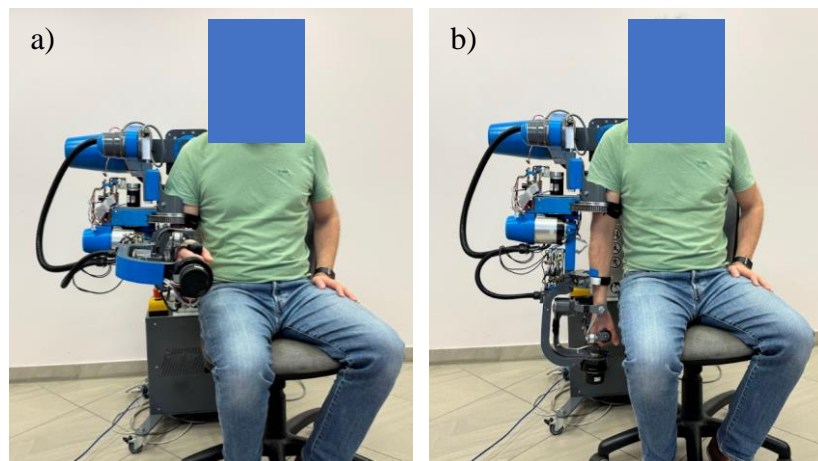
Rys. 5.22 Węzeł 6 - zgięcie dłoniowo-grzbietowe



Rys. 5.23 Węzeł 7 - rotacja w stawie łokciowym

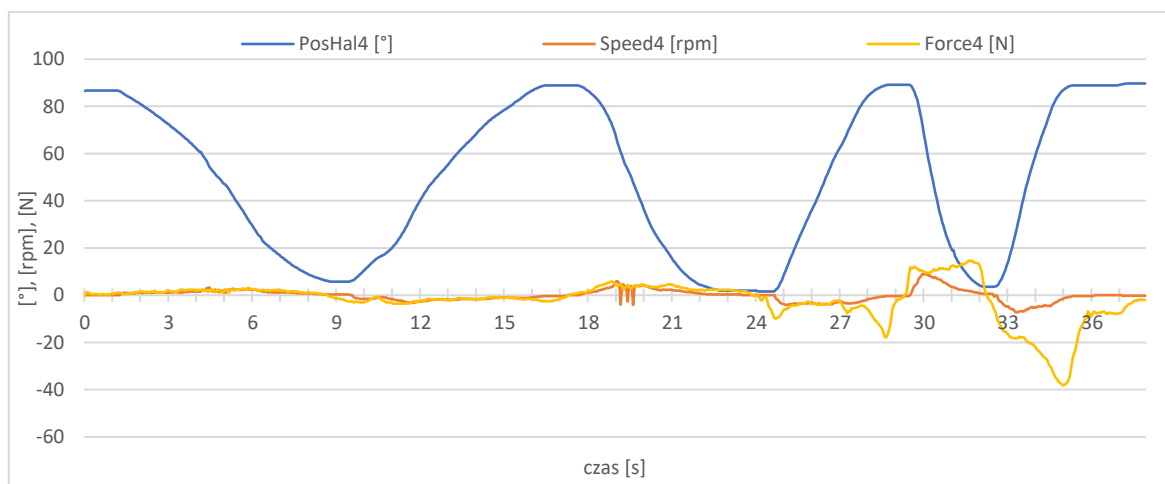
5.6. Badanie węzła robota ARM-200 w trybie śledzenia ruchu.

Celem badania było potwierdzenie możliwości śledzenia ruchu kończyny wpiętej do ramienia prototypu robota rehabilitacyjnego ARM-200. Podczas badania, analizowana była prędkość ruchu ramienia robota, której wartość była zależna od przyłożonej siły przez przedramię śledzonej kończyny. Badaniu poddano węzeł nr 4 robota, odpowiedzialny za zgięcie w stawie łokciowym. Podczas badania, robot został ustawiony w tryb śledzenia ruchu. Pozycją początkową kończyny było zgięcie w stawie łokciowym pod kątem 90° (Rys. 5.24a), następnie staw łokciowy był prostowany do pozycji 0° (Rys. 5.24b). Wykonano trzy pełne cykle prostowania i zginania stawu łokciowego, zadając w każdym cyklu większą siłę nacisku.



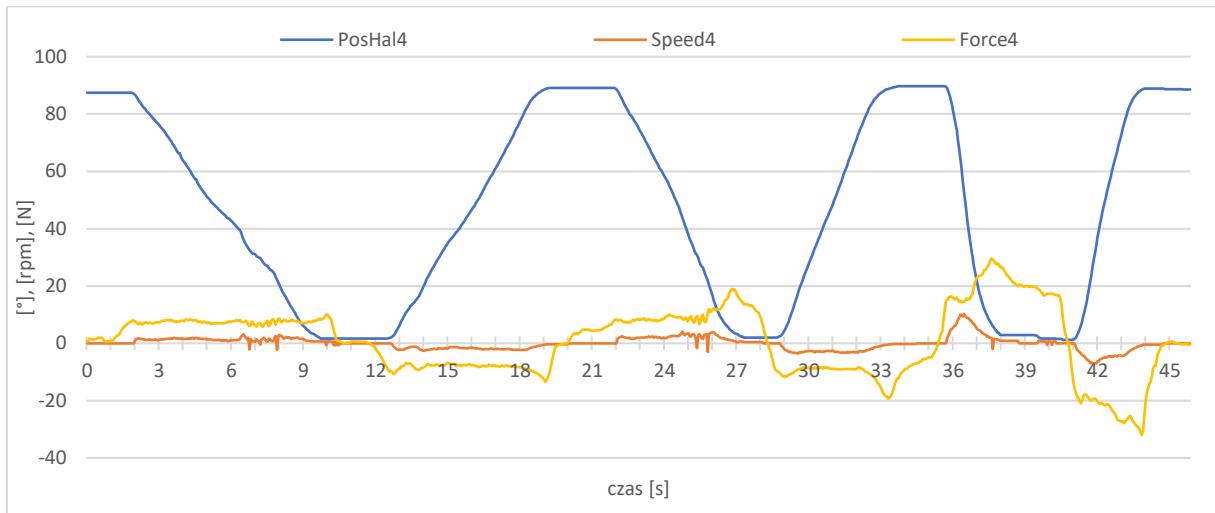
Rys. 5.24 Ustawienie stawu łokciowego: a) zgięcie 90° , b) wyprost 0°

Parametry węzła na Rys. 5.25 przedstawiają zachowanie węzła nr 4 robota ARM-200 w trybie śledzenia ruchu, przy wyłączonych progach niezczułości siły. Jak widać na wykresach, niewielka siła pozwala uruchomić ruch ramienia robota. Tak wysoka czułość układu prowadziła



Rys. 5.25 Badanie węzła nr 4 robota ARM-200 z wyłączonymi progami niezczułości siły

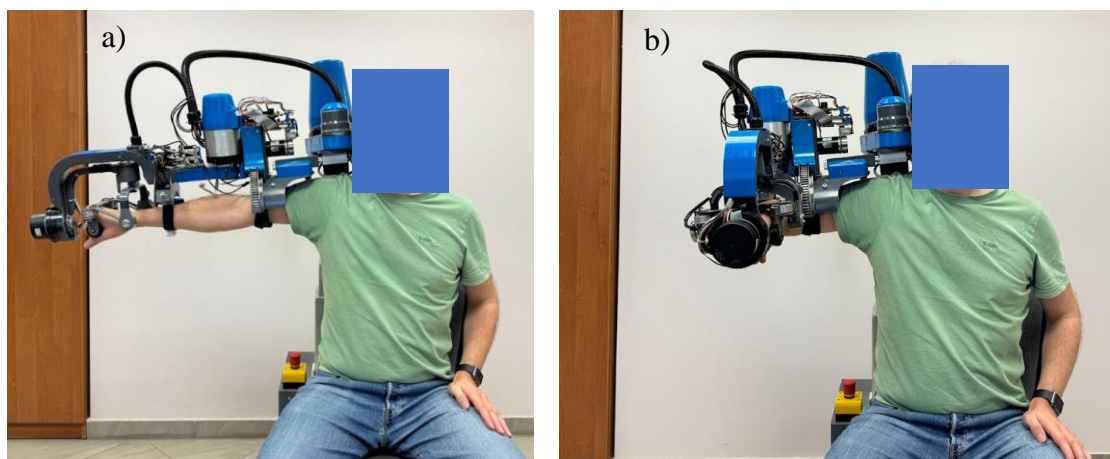
do nieintencjonalnego wprawiania w ruch napędów robota. Problem został rozwiązany przez wprowadzenie parametru określającego stopień niezręczności układu pomiarowego siły. Rys. 5.26 przedstawia parametry węzła nr 4 robota ARM-200, podczas powtórnego eksperymentu z ustawionym progiem niezręczności na wartość 3 N. Jak widać na wykresie, po wprowadzeniu progu niezręczności, podobną szybkość ramienia robota uzyskuje się przy wyższych siłach nacisku, gdzie zmiana amplitudy jest wyraźnie widoczna.



Rys. 5.26 Badanie węzła nr 4 robota ARM-200 z ustawionym progami niezręczności siły na 3 N

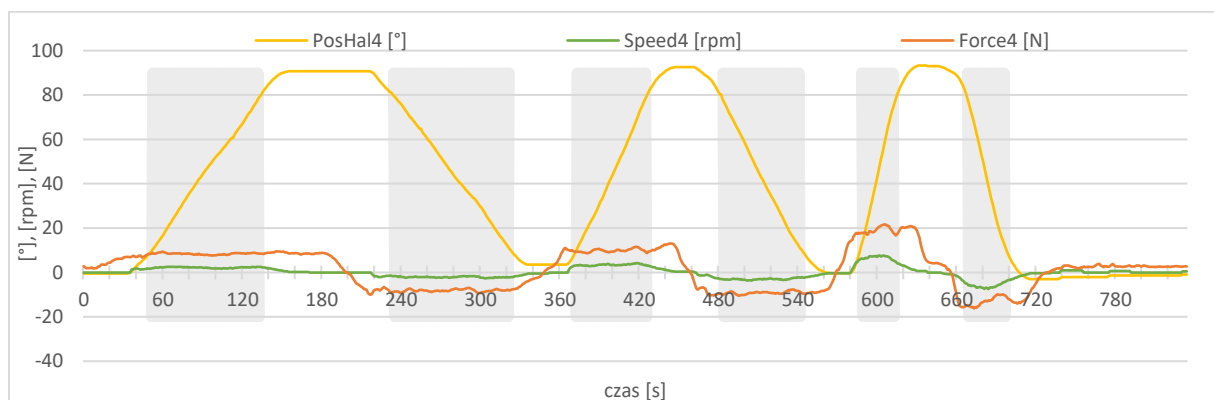
5.7. Badanie ilościowe węzła robota ARM-200 w trybie śledzenia ruchu

Celem kolejnego badania była walidacja algorytmu sterowania napędami robota pracującego w trybie śledzenia ruchu. W badaniu tym sprawdzano zależność wywieranej siły na uzyskaną prędkość ruchu ramienia robota. Badaniu poddano węzeł nr 4 robota, odpowiedzialny za zgięcie w stawie łokciowym. By zredukować potencjalny wpływ sił grawitacji na prędkość ruchu kończyny, badanie przeprowadzono w pozycji, w której kończynę górną odwiedziono do około 90° . Podczas badania, robot został ustawiony w trybie śledzenia ruchu z progiem nieczułości ustawionym na wartość 6 N. Pozycją początkową kończyny był wyprostowany staw łokciowy (Rys. 5.27a), następnie staw łokciowy był zginany do pozycji 90° (Rys. 5.27b). Wykonano trzy pełne cykle prostowania i zginania stawu łokciowego, zadając w każdym cyklu większą siłę nacisku.



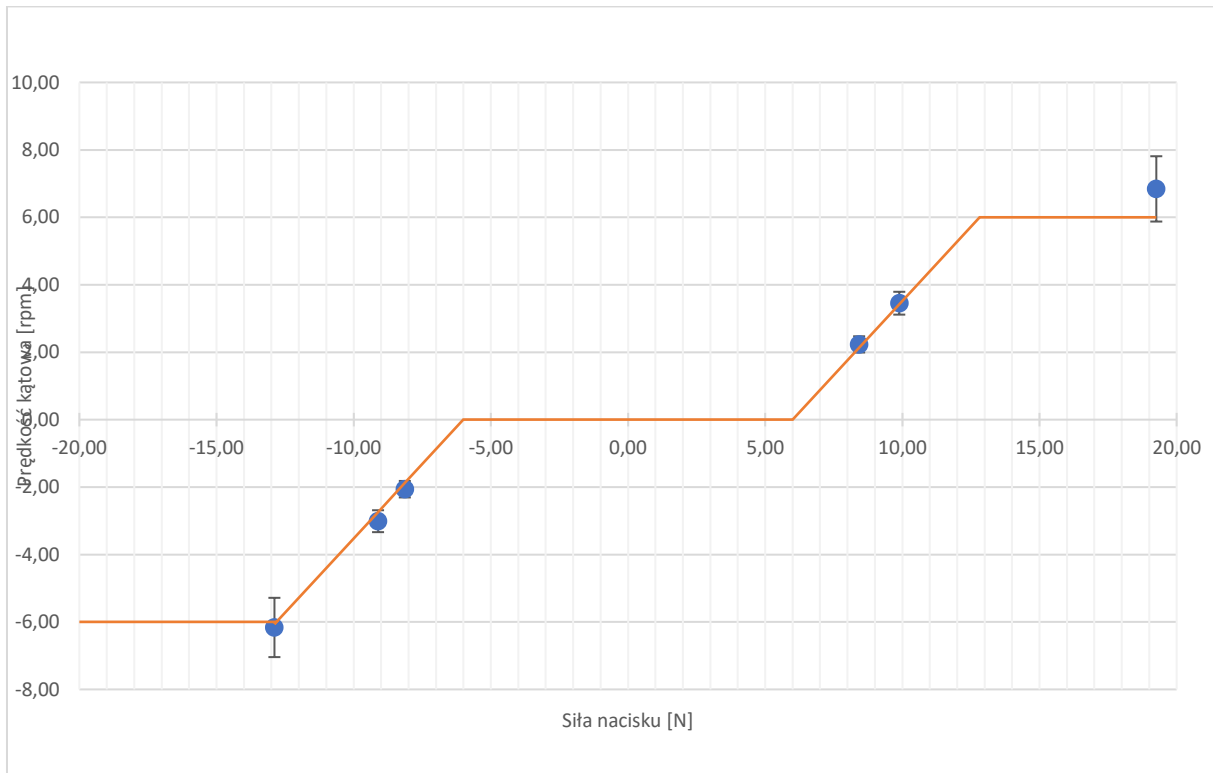
Rys. 5.27 Badanie węzła nr 4 robota ARM-200 w odwiedzeniu kończyny

Analizę uzyskanych prędkości przeprowadzono w przedziale kątowym ramienia robota od 10° do 80° , które zostały na Rys. 5.28 zaznaczone szarym tłem.



Rys. 5.28 Badanie prędkości ruchu ramienia robota ARM-200 w węźle nr 4

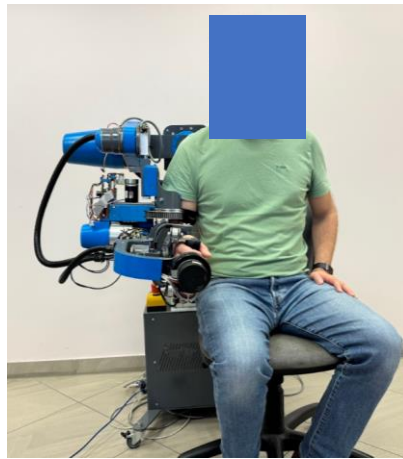
Wykres na Rys. 5.29 przedstawia zmierzone prędkości ruchu dla przyłożonych podczas eksperymentu sił. Dodatkowo, linia czerwona przedstawia oczekiwaną odpowiedź układu sterowania z obszarem nieczułości dla sił w przedziale ± 6 N i ograniczeniem prędkości do 6 obr./min. Jak widać na wykresie, uzyskane prędkości znajdują się w pobliżu założonych wartości prędkości, co potwierdza poprawność działania opracowanego układu sterowania pracującego w trybie śledzenia ruchu.



Rys. 5.29 Pomiar prędkości w zależności od przyłożonej siły, kolorem czerwonym zaznaczono zakładaną odpowiedź układu sterowania

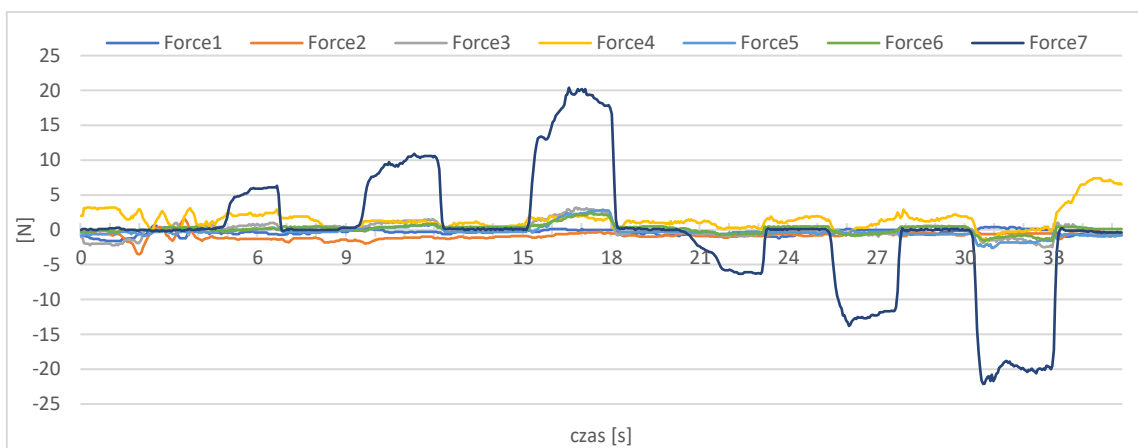
5.8. Wyznaczenie progów nieczułości sił dla robota ARM-200

Na podstawie przeprowadzonych eksperymentów, można zauważyć istotną rolę progów nieczułości torów pomiarowych siły robota. By określić wymagane ustawienia progów nieczułości dla wszystkich węzłów robota przeprowadzono kolejny eksperyment, w którym badano interakcję czujników sił pomiędzy sobą. W eksperymencie, zostały unieruchomione napędy robota, a następnie wywierano siły zgodnie z przyjętym schematem, kolejno na każdy czujnik siły, jednocześnie analizując odpowiedzi czujników robota, na które nie była wywierana intencjonalnie siła. Podczas eksperymentu, ramię robota było ustawione w pozycji przedstawionej na Rys. 5.30.



Rys. 5.30 Ustawienie ramienia robota ARM-200 podczas badania

Podczas badania starano się wywierać kolejno siły: 5 N, 10 N, 20 N, -5 N, -10 N, -20 N. Rys. 5.31 prezentuje przykład odpowiedzi czujników robota ARM-200, podczas wywierania siły na czujnik nr 7.



Rys. 5.31 Odpowiedzi czujników sił robota ARM-200, przy wywieraniu siły na czujnik nr 7

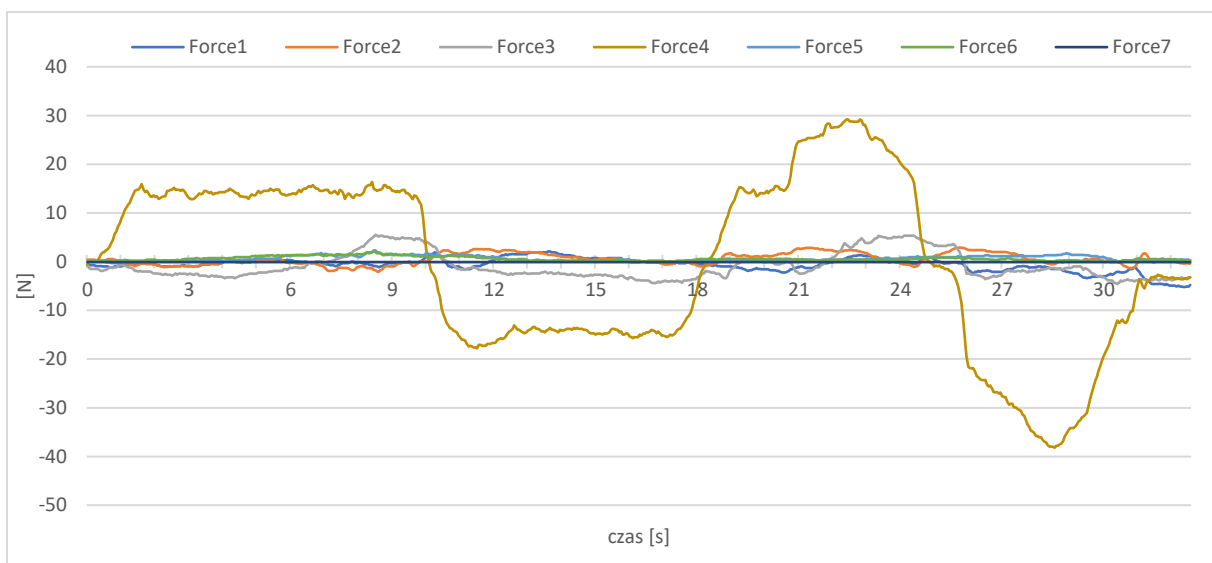
Na podstawie przeprowadzonych badań dla każdego węzła robota zebrano w tabeli Tab. 5.6 moduły wartości maksymalnych sił mierzonych przez węzły, na które nie wywierano w danym etapie badania siły. Dolny wiersz tabeli prezentuje wartości maksymalne, występujące w danym węźle robota. Na podstawie tych wartości wyznaczono wartość progową, wynoszącą 6,4 N, powyżej której napęd robota może przejść do realizacji ruchu, jeżeli robot pracuje w trybie śledzenia ruchu.

Tab. 5.6 Wyniki badania nieintencjonalnej aktywności czujników sił robota ARM-200

		Wartość maksymalna modułu wartości siły dla nacisku $-20N \div 20N$						
Nr węzła:		1.	2.	3.	4.	5.	6.	7.
		[N]	[N]	[N]	[N]	[N]	[N]	[N]
Wybór badanego węzła	1.		4,60	5,70	3,90	1,40	1,70	0,70
	2.	6,10		3,90	4,80	0,80	0,50	0,10
	3.	2,10	2,70		5,10	0,60	1,10	0,70
	4.	2,40	2,30	4,80		1,60	0,50	0,60
	5.	0,60	2,50	1,20	0,80		1,50	6,40
	6.	2,40	1,60	3,10	1,20	1,60		1,60
	7.	0,40	1,20	1,50	1,30	0,80	0,80	
Max:		6,10	4,60	5,70	5,10	1,60	1,70	6,40

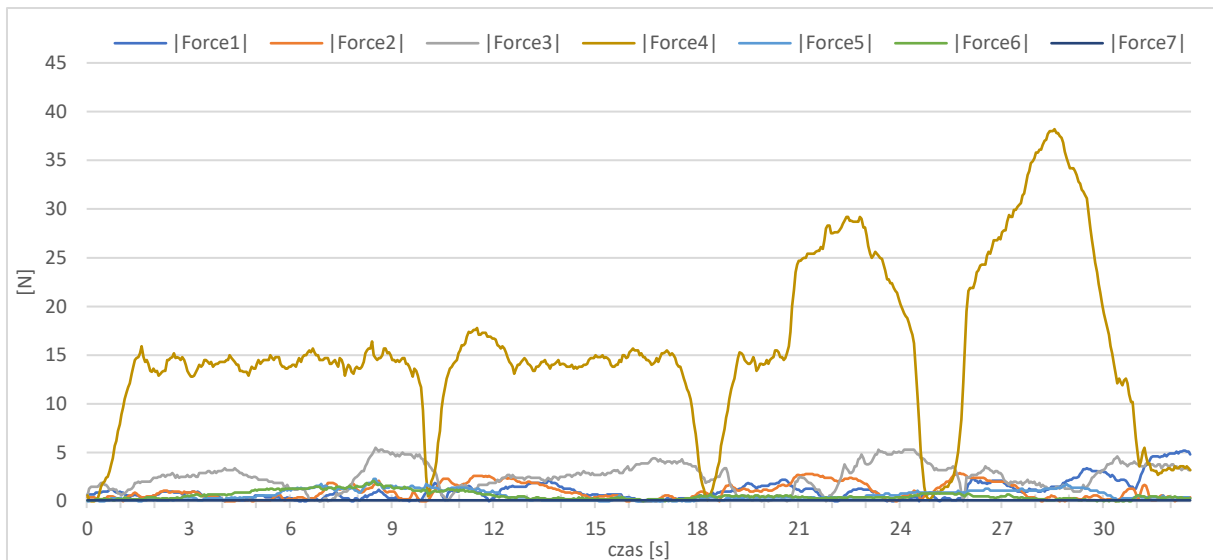
5.9. Walidacja trybu śledzenia ruchu z progami nieczułości

Po wyznaczeniu progu nieczułości dla robota na poziomie 6,4 N i ustawieniu takiego progu dla wszystkich węzłów w aplikacji sterującej pracą robota rehabilitacyjnego, przeprowadzono badania walidujące algorytm sterowania pracą robota w trybie śledzenia ruchu z ustawionymi progami nieczułości. Badanie to polegało na wykonaniu dwóch serii ruchu zginania i prostowania w stawie łokciowym (węzeł nr 4) z różnymi prędkościami ruchu. Zarejestrowane siły działające na poszczególne węzły robota, podczas badania przedstawiono na Rys. 5.32. W badaniu tym sprawdzano wartości prędkości w poszczególnych węzłach.



Rys. 5.32 Wykres wartości sił w węzłach robota, podczas zginanie w stawie łokciowym

Z uwagi na fakt, iż próg nieczułości dotyczy wartości bezwzględnej siły, wyliczono moduły z uzyskanych wartości sił nacisku w węzłach robota, co prezentuje Rys. 5.33.

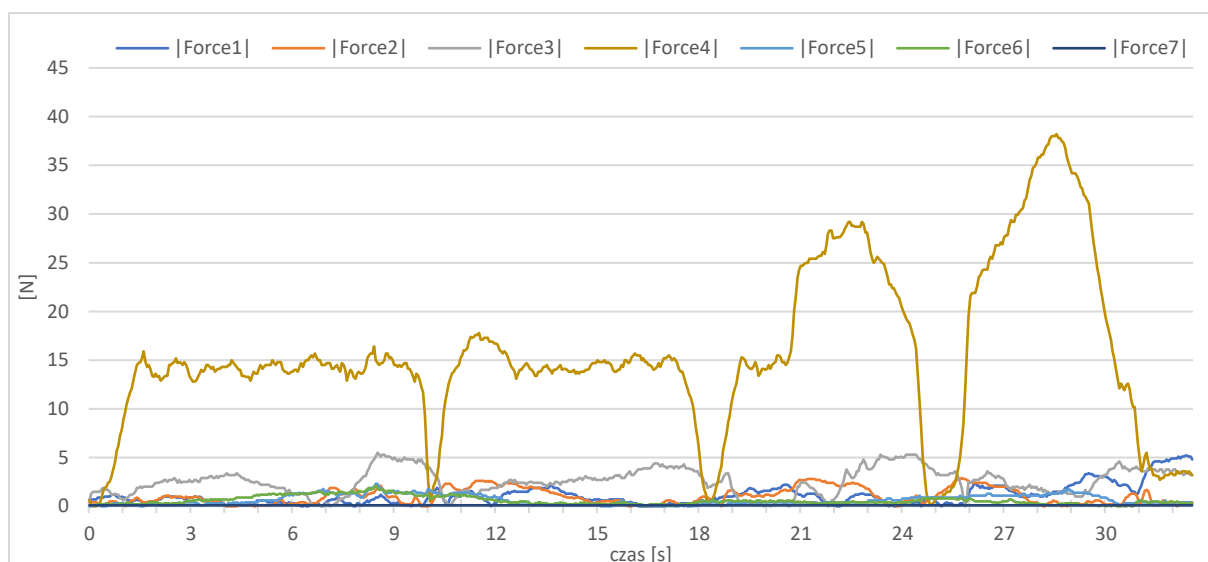


Rys. 5.33 Wykres modułów wartości sił w węzłach robota, podczas zginanie w stawie łokciowym

Tab. 5.7 Wartości sił i prędkości zarejestrowanych podczas badania z progiem niezręczności 6,4 N

	Wartość średnia modułu siły	Wartość maksymalna modułu siły	Wartość średnia modułu prędkości	Wartość maksymalna modułu prędkości
Nr węzła:	[N]	[N]	[rpm]	[rpm]
1.	0,98	5,20	0,00	0,00
2.	0,90	2,90	0,00	0,00
3.	2,18	5,50	0,00	0,00
4.	13,56	38,20	1,23	4,40
5.	0,61	2,30	0,00	0,00
6.	0,51	2,10	0,00	0,00
7.	0,08	0,10	0,00	0,00

Z uwagi na fakt, iż próg niezręczności dotyczy wartości bezwzględnej siły, wyliczono moduły z uzyskanych wartości sił nacisku w węzłach robota, co prezentuje Rys. 5.33.



Rys. 5.33 Wykres modułów wartości sił w węzłach robota, podczas zginanie w stawie łokciowym

Tab. 5.7 zestawiono wartości średnie oraz maksymalne wyznaczone z przebiegów sił i prędkości kątowych zarejestrowanych w poszczególnych węzłach robota podczas badania. Analiza danych zestawionych w tabeli jednoznacznie wskazuje, że w żadnym węźle oprócz węzła 4, nie został przekroczony próg nieczułości. Co ważne w każdym węźle, oprócz węzła napędzanego węzła nr 4, wartość średnia i maksymalna modułu prędkości kątowej była równa zero. Oznacza to prawidłowe działanie układu sterowania w trybie śledzenia ruchu. Wprowadzone progi nieczułości zapewniły brak ruchu w węzłach nieaktywnych. Żaden z węzłów robota oprócz aktywnego węzła nr 4, nie przekroczył poziomu siły aktywującej napęd robota.

6. PODSUMOWANIE I WNIOSKI

Celem doktoratu wdrożeniowego było opracowanie modułowego systemu sterowania robota rehabilitacyjnego przeznaczonego do terapii kończyn górnych wykorzystującego sprzężenie siłowe do realizacji ruchów robota.

Efektom zrealizowanych prac projektowych i wdrożeniowych było opracowanie układu sterowania robota rehabilitacyjnego do terapii kończyn górnych, w szczególności:

- zaprojektowanie i wykonanie elementów sprzętowych systemu sterowania robota rehabilitacyjnego,

- opracowanie i implementacja algorytmów sterowania systemu sterowania robota rehabilitacyjnego,
- opracowanie protokołów komunikacji pomiędzy elementami systemu sterowania robota rehabilitacyjnego,
- opracowanie metodyki doboru nastaw regulatorów PID elementów wykonawczych systemu sterowania robota rehabilitacyjnego,

Zaprojektowany i wykonany prototyp robota rehabilitacyjnego został wykorzystany w pracach badawczych, których głównym celem była walidacja opracowanego systemu sterowania, obejmująca:

- walidację układów pomiarowych w zaprojektowanym systemie sterowania, badanie te dotyczyły w szczególności oceny poprawności działania czujników siły, czujników położenia kąтового, czujników prędkości kątovej,
- ocenę potencjalnego negatywnego wpływu zastosowania do sterowania pracą systemu oprogramowania działającego w systemie nie czasu rzeczywistego,
- walidację algorytmów sterowania w tzw. „trybie śledzenia ruchu” wykorzystującego sprzężenie siłowe oraz algorytmy automatycznej detekcji intencji ruchu w stawach kończyny górnej pacjenta,
- przeprowadzenie badań pozwalających na wyznaczenie tzw. „progów nieczułości” - minimalnych sił wymaganych do aktywacji napędów poszczególnych węzłów robota zapewniających eliminację niepożądanych ruchów robota,
- przeprowadzenie badań walidujących opracowany system sterowania w „trybie śledzenia ruchu” z uwzględnieniem wyznaczonych progów nieczułości.

Uzyskane wyniki przeprowadzonych badań walidacyjnych elementów układu sterowania robota rehabilitacyjnego kończyn górnych potwierdziły poprawność ich działania oraz spełnienie wymagań funkcjonalnych i normatywnych zaprojektowanego i wykonanego prototypu robota rehabilitacyjnego.

Na podstawie przeprowadzonych badań doświadczalnych potwierdzono również tezy pracy:

4. Dane z czujników pomiarowych siły umieszczonych w poszczególnych węzłach robota rehabilitacyjnego można wykorzystać do automatycznej detekcji intencji ruchu w stawach kończyny górnej pacjenta oraz do sterowania trajektorią ruchu robota.
5. Wprowadzenie w algorytmie sterowania ruchami robota minimalnej siły wymaganej do aktywacji napędów przyczyni się do zwiększenia bezpieczeństwa pacjenta poprzez wyeliminowanie niepożądanych ruchów robota.
6. Wykorzystanie aplikacji sterującej działającej w systemie operacyjnym nie czasu rzeczywistego (ang.: non-real-time system), do kontroli pracy robota rehabilitacyjnego, nie wpłynie negatywnie na pracę systemu sterowania robota.

7. KIERUNKI DALSZYCH BADAŃ

Przeprowadzone badania i uzyskane doświadczenia z pracy systemu sterowania robota na stanowisku badawczym pokazują, że pomimo uzyskania zadowalających efektów pracy opracowanego systemu sterowania można wskazać główne potrzeby dalszego doskonalenia systemu sterowania:

- Umożliwienie realizacji trybu śledzenia ruchu robota z wykorzystaniem mniejszych progów nieczułości.
- Dodanie możliwości kontroli impedancji mechanicznej na styku człowiek-robot.
- Opracowanie rozwiązania o większej miniaturyzacji pakietów elektronicznych szczególnie modułów Slave rozmieszczonych na konstrukcji mechanicznej robota.
- Opracowanie własnej platformy 3D do realizacji gier rehabilitacyjnych.
- Opracowanie algorytmów sterowania robota uwzględniających sygnały z torów pomiarowych sEMG.

STRESZCZENIE

STRESZCZENIE ROZPRAWY DOKTORSKIEJ

„Modularny system sterowania robota rehabilitacyjnego narządów ruchu ze sprzężeniem siłowym”

Rehabilitacja kończyn, mająca na celu przywrócenie utraconej sprawności, jest procesem długotrwałym. W tym procesie, fizjoterapeuta podczas zazwyczaj wielomiesięcznych ćwiczeń, wymagających dużego wysiłku fizycznego, wykonuje manualnie ruchy rehabilitowaną kończyną pacjenta. By ograniczyć zmęczenie rehabilitanta i jednocześnie zapewnić większą powtarzalność aplikowanych ćwiczeń terapeutycznych zaczęto wprowadzać różnego rodzaju urządzenia mechaniczne, jak również mechatroniczne, które aktualnie w najbardziej zaawansowanej formie przybrały postać robotów rehabilitacyjnych.

Przedmiotem niniejszej pracy było zaprojektowanie i walidacja systemu sterowania dla robota rehabilitacyjnego o wielu stopniach swobody, w którym do kontroli pracy robota zostało wykorzystane sprzężenie siłowe. Podjęcie takiej tematyki badań wynikało z prac prowadzonych w Instytucie Techniki i Aparatury Medycznej ITAM w obszarze projektowania i wdrażania robotów rehabilitacyjnych. Wynikiem prac badawczych i projektowych prowadzonych w ITAM w latach 2007÷2009 był prototyp robota rehabilitacyjnego ARM-100. Robot ten w formie egzoszkieletu umożliwiał wykonywanie wielopłaszczyznowych ruchów terapeutycznych.

Na podstawie szerokiej gamy badań testujących i walidujących prototyp robota rehabilitacyjnego ARM-100 określono kierunki dalszego rozwoju opracowanego rozwiązania. Jednym z istotniejszych wniosków było wskazanie konieczności opracowania nowego systemu sterowania, który na podstawie danych z czujników pomiarowych umożliwiałby określenie intencji ruchu w stawach, a elementy napędowe robota wspomagałyby ten ruch. W 2017 r. w ITAM rozpoczęły się prace związane z opracowaniem kolejnej udoskonalonej wersji robota rehabilitacyjnego kończyn górnych ARM-200, a realizacja niniejszego doktoratu wdrożeniowego była ściśle powiązana z tymi pracami, przede wszystkim w zakresie opracowania systemu sterowania. W toku prac projektowych związanych z opracowaniem

nowej wersji robota określone zostały wymagania funkcjonalne i normatywne, które były podstawą do sformułowania celu pracy, który ukierunkowany była na opracowanie i walidację systemu sterowania robota rehabilitacyjnego przeznaczonego do terapii kończyn górnych wykorzystującego sprzężenie siłowe do realizacji ruchów robota. Zakres prac badawczych zrealizowanych w rozprawie doktorskiej został doprecyzowany przyjętymi tezami badawczymi w brzmieniu:

1. Dane z czujników pomiarowych siły umieszczonych w poszczególnych węzłach robota rehabilitacyjnego można wykorzystać do automatycznej detekcji intencji ruchu w stawach kończyny górnej pacjenta oraz do sterowania trajektorią ruchu robota.
2. Wprowadzenie w algorytmie sterowania ruchami robota minimalnej siły wymaganej do aktywacji napędów (tzw. progów nieczułości) przyczyni się do zwiększenia bezpieczeństwa pacjenta poprzez wyeliminowanie niepożądanych ruchów robota.
3. Wykorzystanie aplikacji sterującej działającej w systemie operacyjnym nie czasu rzeczywistego (ang.: non-real-time system), do kontroli pracy robota rehabilitacyjnego, nie wpłynie negatywnie na pracę systemu sterowania robota.

Finalnym efektem zrealizowanych prac projektowych i wdrożeniowych było opracowanie układu sterowania robota rehabilitacyjnego do terapii kończyn górnych. Istotnym elementem tego układu sterowania było opracowanie algorytmów sterowania umożliwiających automatyczną detekcję intencji ruchu w stawie oraz wspomaganie tego ruchu. Zaprojektowany i wykonany prototyp robota rehabilitacyjnego został wykorzystany w pracach badawczych, których głównym celem była walidacja opracowanego systemu sterowania.

Uzyskane wyniki przeprowadzonych badań walidacyjnych elementów układu sterowania robota rehabilitacyjnego kończyn górnych potwierdziły poprawność ich działania oraz spełnienie wymagań funkcjonalnych i normatywnych zaprojektowanego i wykonanego prototypu robota rehabilitacyjnego. Na podstawie przeprowadzonych badań doświadczalnych potwierdzono również tezy pracy.

SUMMARY

SUMMARY OF THE DOCTORAL THESIS

"Modular control system for a musculoskeletal rehabilitation robot with force feedback"

Limb rehabilitation to restore lost function is a long-term process. In this process, the physiotherapist manually moves the rehabilitated limb of the patient during exercises that usually last for months and require a lot of physical effort. To reduce the fatigue of the physiotherapist and, at the same time, ensure greater repeatability of the applied therapeutic exercises, various types of mechanical and mechatronic devices began to be introduced, which currently, in their most advanced form, have taken the form of rehabilitation robots.

The subject of this study was the design and validation of a control system for a rehabilitation robot with multiple degrees of freedom, in which force feedback was used to control the robot. Taking up such a research topic resulted from the work carried out at the Institute of Medical Technology and Equipment ITAM in the area of rehabilitation robot design and implementation. The result of the research and design work carried out at ITAM in 2007÷2009 was the prototype of the ARM-100 rehabilitation robot. This robot, in the form of an exoskeleton, enabled the performance of multi-planar therapeutic movements.

On the basis of a wide range of testing and validation studies of the ARM-100 rehabilitation robot prototype, directions for further development of the developed solution were identified. One of the most important conclusions was the need to develop a new control system, which, based on the data from the measurement sensors, would be able to determine the intention of movement in the joints, and the robot's drive components would support this movement. In 2017, work began at ITAM to develop a further improved version of the ARM-200 upper limb rehabilitation robot, and the execution of this implementation PhD was closely linked to this work, primarily in the development of the control system. In the course of the design work related to the development of the new version of the robot, functional and normative requirements were identified, which formed the basis for the formulation of the aim of

the thesis, which was directed towards the development and validation of a control system for a rehabilitation robot designed for upper limb therapy using force coupling to realise the robot's movements. The scope of the research work carried out in the dissertation was refined by the adopted research theses as follows:

- 1 The data from the force measurement sensors placed at the individual nodes of the rehabilitation robot can be used to automatically detect movement intentions in the patient's upper limb joints and to control the robot's movement trajectory.
2. The introduction in the robot's movement control algorithm of the minimum force required to activate the actuators (so-called insensitivity thresholds) will contribute to increasing patient safety by eliminating unwanted robot movements.
3. The use of a control application operating in a non-real-time system to control the operation of the rehabilitation robot will not adversely affect the operation of the robot control system.

The final result of the completed design and implementation work was the development of a control system for a rehabilitation robot for upper limb therapy. An important element of this control system was the development of control algorithms to enable automatic detection of joint movement intentions and to support this movement. The designed and manufactured prototype of the rehabilitation robot was used in research work, the main aim of which was to validate the developed control system.

The results obtained from the validation tests of the control system components of the upper limb rehabilitation robot confirmed the correctness of their operation and the fulfilment of the functional and normative requirements of the designed and manufactured rehabilitation robot prototype. The theses of the study were also confirmed on the basis of the conducted experimental tests.

LITERATURA

- [1] D. Szałtys, Red., *Starzenie się ludności Polski w świetle wyników narodowego spisu powszechnego ludności i mieszkań 2021*. Warszawa: Główny Urząd Statystyczny, Departament Badań Demograficznych, 2023. [Online]. Dostępne na: https://stat.gov.pl/download/gfx/portalinformacyjny/pl/defaultaktualnosci/6536/3/1/1/starzenie_sie_ludnosci_polski_nsp_2021.pdf
- [2] „INFORMACJA Rządu Rzeczypospolitej Polskiej o działaniach podejmowanych w 2021r. na rzecz realizacji postanowień uchwały Sejmu Rzeczypospolitej Polskiej z dnia 1 sierpnia 1997 r. KARTA PRAW OSÓB NIEPEŁNOSPRAWNYCH”, Warszawa, 2022. [Online]. Dostępne na: <https://www.sejm.gov.pl/Sejm9.nsf/druk.xsp?nr=2483>
- [3] J. Janik, „Egzoszkielety, roboty rehabilitacyjne - to już się dzieje. Przed nami przyszłość rehabilitacji medycznej”, 13 grudzień 2021. [Online]. Dostępne na: <https://www.rynekzdrowia.pl/Uslugi-medyczne/Egzoszkielety-roboty-rehabilitacyjne-to-juz-sie-dzieje-Przed-nami-przyszlosc-rehabilitacji-medycznej,227602,8.html>
- [4] J. Sobotta, *Atlas anatomii człowieka Sobotta Ogólne pojęcia anatomiczne Narządy ruch*, 24. wyd., t. I. Wrocław: Edra Urban & Partner, 2019.
- [5] M. Żak, „Raport na temat przyczyn i następstw upadków osób w wieku senioralnym”, Kraków, 2017. [Online]. Dostępne na: https://fundacijaneuca.pl/wp-content/uploads/2023/05/M.ZAK_Raport_upadki-osob-starszych_01.08.2017.pdf
- [6] „NFZ o zdrowiu. Udar niedokrwienny mózgu”, Warszawa, 2024. [Online]. Dostępne na: <https://ezdrowie.gov.pl/portal/home/badania-i-dane/zdrowe-dane/raporty/nfz-o-zdrowiu-udar-niedokrwienny-mozgu>
- [7] P. Szermer i A. Szczudlik, „Udar: zasady postępowania”, w *Udar mózgu*, Kraków: Wydawnictwo Uniwersytetu Jagiellońskiego, 2007, s. 234–241. [Online]. Dostępne na: <https://ruj.uj.edu.pl/bitstreams/ca9b4e44-2caf-4e4b-9238-b0ea4a633cbf/download>
- [8] M. Paprocka-Borowicz i M. Zawadzki, *Fizjoterapia w chorobach układu ruchu*. Wrocław: Górnicki Wydawnictwo Medyczne, 2019.
- [9] R. Horst, *Trening strategii motorycznych i PNF*. Kraków: Top School s.c., 2010.
- [10] S. S. Adler, D. Beckers, i M. Buck, *PNF w Praktyce Ilustrowany Przewodnik*, IV. DB PUBLISHING, 2014.
- [11] R. Michnik i in., *Kinematic Analysis of Complex Therapeutic Movements of the Upper Limb*. w *Information Technologies in Biomedicine*, no. 47. Springer–Verlag Berlin Heidelberg, 2008.
- [12] I. Díaz, J. J. Gil, i E. J. Sánchez, „Lower-Limb Robotic Rehabilitation: Literature Review and Challenges”, *Journal of Robotics*, 2011. doi: 10.1155/2011/759764.
- [13] E. Koutsiana, I. Ladakis, D. Fotopoulos, A. Chytas, V. Kilintzis, i I. Chouvarda, „Serious Gaming Technology in Upper Extremity Rehabilitation: Scoping Review”, 2020, doi: 10.2196/19071.
- [14] „Gustave Zander and the 19th-Century Gym — The Public Domain Review”. Dostęp: 14 lipiec 2024. [Online]. Dostępne na: <https://publicdomainreview.org/collection/zander-gym/>
- [15] J. Hidler, D. Nichols, M. Pelliccio, i K. Brady, „Advances in the Understanding and Treatment of Stroke Impairment Using Robotic Devices”, *Top. Stroke Rehabil.*, nr 12(2), s. 22–35, 2005, doi: 10.1310/RYT5-62N4-CTVX-8JTE.
- [16] H. I. Krebs, N. Hogan, M. L. Aisen, i B. T. Volpe, „Robot-aided neurorehabilitation”, *IEEE Trans Rehabil Eng*, nr 6(1), s. 75–87, 1998, doi: 10.1109/86.662623.
- [17] J. Dunaj, W. J. Klimasara, i Z. Pilat, „Human-Robot Interaction in the Rehabilitation Robot Renus-1”, w *Recent Advances in Systems, Control and Information Technology*, R.

- Szewczyk i M. Kaliczyńska, Red., Cham: Springer International Publishing, 2017, s. 358–367. doi: 10.1007/978-3-319-48923-0_39.
- [18] J. Dunaj i W. Klimasara, „Hardware and software solutions in the control of rehabilitation robots Renus”, *Pomiary Autom. Robot.*, t. 18, nr 12, s. 100–115, grudz. 2014, doi: 10.14313/PAR_214/100.
- [19] Z. Pilat, W. J. Klimasara, Ł. Juszyński, i A. Michnik, „Research and Development of Rehabilitation Robotics in Poland”, *Appl. Mech. Mater.*, t. 613, s. 196–207, sie. 2014, doi: 10.4028/www.scientific.net/AMM.613.196.
- [20] M. Mihelj, T. Nef, i R. Riener, „ARMin II - 7 DoF rehabilitation robot: mechanics and kinematics”, w *Proceedings 2007 IEEE International Conference on Robotics and Automation*, Rome, Italy: IEEE, kwi. 2007, s. 4120–4125. doi: 10.1109/ROBOT.2007.364112.
- [21] M. Babaiasl, S. H. Mahdioun, P. Jaryani, i M. Yazdani, „A review of technological and clinical aspects of robot-aided rehabilitation of upper-extremity after stroke”, *Disabil. Rehabil. Assist. Technol.*, s. 1–18, sty. 2015, doi: 10.3109/17483107.2014.1002539.
- [22] „Technical Data Sheet - Hocoma”. Dostęp: 14 lipiec 2024. [Online]. Dostępne na: <https://www.hocoma.com/us/solutions/armeo-power/technical-data-sheet/>
- [23] S. Taravati, K. Capaci, H. Uzumcugil, i G. Tanigor, „Evaluation of an upper limb robotic rehabilitation program on motor functions, quality of life, cognition, and emotional status in patients with stroke: a randomized controlled study”, *Neurol. Sci.*, t. 43, nr 2, s. 1177–1188, luty 2022, doi: 10.1007/s10072-021-05431-8.
- [24] „Rehab Robot HARMONY - ReNeu Robotics Lab, UT Austin Medical Exoskeleton Report”. Dostęp: 14 lipiec 2024. [Online]. Dostępne na: <https://exoskeletonreport.com/2015/05/rehab-robot-harmony-reneu-robotics-lab-ut-austin/>
- [25] „Bilateral Arm-Shoulder-Therapy | DIEGO®”. Dostęp: 14 lipiec 2024. [Online]. Dostępne na: <https://tyromotion.com/en/products/diego/>
- [26] „Samcon - Luna EMG”. Dostęp: 15 lipiec 2024. [Online]. Dostępne na: <https://samcon.be/Rehabilitation/Robotics/Luna-EMG>
- [27] „EGZOTech_katalog_2022”. Dostęp: 14 lipiec 2024. [Online]. Dostępne na: https://doc.egzotech.com/EGZOTech_katalog_2022_podglad_rozkadowka.pdf
- [28] A. Michnik i in., „Rehabilitation Robot Prototypes Developed by the ITAM Zabrze”, *Arch. Mech. Eng.*, t. 61, nr 3, s. 433–444, grudz. 2014, doi: 10.2478/meceng-2014-0024.
- [29] *Film prezentujący pracę prototypu robota ARM-100*, (2015). Dostęp: 22 lipiec 2024. [Online Video]. Dostępne na: https://www.youtube.com/watch?v=BP1_qn9Ilog
- [30] L. Masia, H. I. Krebs, P. Cappa, i N. Hogan, „Design and Characterization of Hand Module for Whole-Arm Rehabilitation Following Stroke”, *IEEEASME Trans. Mechatron.*, t. 12, nr 4, s. 399–407, sie. 2007, doi: 10.1109/TMECH.2007.901928.
- [31] R. Tong, *Biomechatronics in Medicine and Healthcare*. Singapur: Pan Stanford Publishing Pte. Ltd., 2011.
- [32] H. M. Qassim i W. Z. Wan Hasan, „A Review on Upper Limb Rehabilitation Robots”, *Appl. Sci.*, t. 10, nr 19, Art. nr 19, sty. 2020, doi: 10.3390/app10196976.
- [33] Q. Meng, Q. Xie, i H. Yu, „Upper-Limb Rehabilitation Robot: State of the Art and Existing Problems”, grudz. 2019. [Online]. Dostępne na: https://www.researchgate.net/publication/338037336_Upper-Limb_Rehabilitation_Robot_State_of_the_Art_and_Existing_Problems
- [34] „International Federation of Robotics”. Dostęp: 29 sierpień 2024. [Online]. Dostępne na: <https://ifr.org/robot-history>
- [35] „ROS/Introduction - ROS Wiki”. Dostęp: 29 sierpień 2024. [Online]. Dostępne na: <http://wiki.ros.org/ROS/Introduction>

- [36] A. Michnik, J. Wołoszyn, M. Sobiech, i G. Nowak, *Upper limb rehabilitation robot*. 2021.
- [37] A. Michnik i in., „Control System for a Limb Rehabilitation Robot”, w *Information Technologies in Biomedicine*, t. 69, E. Piętka i J. Kawa, Red., w *Advances in Intelligent and Soft Computing*, vol. 69. , Berlin, Heidelberg: Springer Berlin Heidelberg, 2010, s. 423–430. doi: 10.1007/978-3-642-13105-9_42.
- [38] M. Sobiech, A. Michnik, I. Chuchnowska, I. Karpiel, i W. Wolański, „Multivariate analysis of the kinematics of an upper limb rehabilitation robot”, nr Volume 26 (2024), s. 55–66, luty 2024, doi: doi.org/10.37190/abb-02386-2024-02.
- [39] „PN-EN 60601-1:2011 - wersja polska”. Dostęp: 13 wrzesień 2024. [Online]. Dostępne na: <https://sklep.pkn.pl/pn-en-60601-1-2011p.html>
- [40] A. Michnik, A. Gacek, Z. Paszenda, J. Brandt, Z. Szczurek, i M. Bachorz, „Urząd Patentowy RP - P.389449 Sposób sterowania napędami egzoszkieletu zwłaszcza do rehabilitacji nadgarstka”. Dostęp: 20 lipiec 2024. [Online]. Dostępne na: <https://ewyszukiwarka.pue.uprp.gov.pl/search/pwp-details/P.389449>
- [41] A. Michnik, A. Gacek, Z. Paszenda, J. Brandt, Z. Szczurek, i M. Bachorz, „Urząd Patentowy RP - P.389451 Układ sterowania napędami egzoszkieletu zwłaszcza do rehabilitacji nadgarstka”. Dostęp: 20 lipiec 2024. [Online]. Dostępne na: <https://ewyszukiwarka.pue.uprp.gov.pl/search/pwp-details/P.389451>
- [42] „ADS1292R data sheet, product information and support | TI.com”. Dostęp: 8 wrzesień 2024. [Online]. Dostępne na: <https://www.ti.com/product/ADS1292R>
- [43] „STM32F051C8 - Mainstream Arm Cortex-M0 Access line MCU with 64 Kbytes of Flash memory, 48 MHz CPU, motor control and CEC functions - STMicroelectronics”. Dostęp: 8 wrzesień 2024. [Online]. Dostępne na: <https://www.st.com/en/microcontrollers-microprocessors/stm32f051c8.html>
- [44] „Nota katalogowa czujnika KMM30”. [Online]. Dostępne na: <https://wobit.com.pl/download/kmm30.pdf>
- [45] W. Styburski, *Przetworniki tensometryczne*, 1. wyd. Warszawa: Wydawnictwa Naukowo-Techniczne, 1976.
- [46] J. Brzózka, *Regulatory i układy automatyki*. Mikom, 2004.
- [47] D. Ibrahim, *PID-based Practical Digital Control with Raspberry Pi and Arduino Uno*, II. Susteren: Elector, 2022.
- [48] H.-B. Shin i J.-G. Park, „Anti-Windup PID Controller With Integral State Predictor for Variable-Speed Motor Drives”, *IEEE Trans. Ind. Electron.*, t. 59, nr 3, s. 1509–1516, mar. 2012, doi: 10.1109/TIE.2011.2163911.
- [49] K. Przystupa, „Tuning of PID Controllers – Approximate Methods”, *Adv. Sci. Technol. Res. J.*, t. 12, nr 4, s. 56–64, grudz. 2018, doi: 10.12913/22998624/99987.
- [50] M. Veronesi i A. Visioli, „On the Selection of Lambda in Lambda Tuning for PI(D) Controllers”, *IFAC-Pap.*, t. 53, nr 2, s. 4599–4604, 2020, doi: 10.1016/j.ifacol.2020.12.486.
- [51] „PID Loop Simulator: Engineers-Excel.com”. Dostęp: 1 wrzesień 2024. [Online]. Dostępne na: https://www.engineers-excel.com/Apps/PID_Simulator/Description.htm
- [52] P. Bogusz, M. Korkosz, i J. Prokop, „Analiza porównawcza właściwości bezszczotkowego silnika z magnesami trwałymi dla różnych konfiguracji uzwojenia stojana - Maszyny Elektryczne : zeszyty problemowe - Tom Nr 2 (118) (2018) - BazTech - Yadda”. Dostęp: 6 sierpień 2024. [Online]. Dostępne na: <https://yadda.icm.edu.pl/baztech/element/bwmeta1.element.baztech-390b14ec-c83e-4af4-864a-9f0290004ebf?printView=true>
- [53] „Online shop for high precise drive systems by maxon | maxon group”. Dostęp: 9 sierpień 2024. [Online]. Dostępne na:

<https://www.maxongroup.com/maxon/view/product/motor/ecmotor/ecflat/ecflat60/408057>

- [54] „maxon Product Range 2024/25”. Dostęp: 7 sierpień 2024. [Online]. Dostępne na: <https://online.flippingbook.com/view/1042987/66/>
- [55] „INA240 data sheet, product information and support | TI.com”. Dostęp: 10 sierpień 2024. [Online]. Dostępne na: <https://www.ti.com/product/INA240>
- [56] K. Krykowski, *Silniki PM BLDC – właściwości, sterowanie, aplikacje*. Legionowo: BTC, 2015.
- [57] A. Domoracki i K. Krykowski, „Silniki BLDC - klasyczne metody sterowania”, *Masz. Elektr. Zesz. Probl.*, t. Nr 72, s. 155–159, 2005.
- [58] K. S. Siyan i T. Parker, *TCP/IP. Księga eksperta*, II. Gliwice: Helion, 2002.
- [59] „SAM4E Series - Atmel | SMART ARM-based Flash MCU - Datasheet”. [Online]. Dostępne na: https://ww1.microchip.com/downloads/aemDocuments/documents/OTH/ProductDocuments/DataSheets/Atmel-11157-32-bit-Cortex-M4-Microcontroller-SAM4E16-SAM4E8_Datasheet.pdf
- [60] H. Osterloh, *TCP/IP. Szkoła programowania*. Gliwice: Helion, 2006.
- [61] K. Krysiak, *Sieci komputerowe: kompendium*. Gliwice: Helion, 2002.
- [62] „Wireshark”. Dostęp: 15 sierpień 2024. [Online]. Dostępne na: <https://www.wireshark.org/>
- [63] M. Kocherovsky, „Accuracy of Timers made in various programming languages”, *Animate the Movement, and Learn Computer Science*, nr ARISE-TM-2019-1, 2019. [Online]. Dostępne na: <https://www.robofest.net/2019/TimerReport.pdf>
- [64] J. Miskovic, „C# Timer: Everything you need to know”. [Online]. Dostępne na: <https://josipmisko.com/posts/c-sharp-timer>

SPIS RYSUNKÓW

Rys. 1.1 Przykłady urządzeń rehabilitacyjnych Wilhelma Zandera [14].....	13
Rys. 1.2 Robot MIT-Manus (Interactive Motion Technologies, Cambridge, MA) [15]	13
Rys. 1.3 Robot RENU-1 (PIAP) [17]	14
Rys. 1.4 Robot ARMin II (ETH Zurich) [20]	15
Rys. 1.5 Robot ARMin III (ETH Zurich) [21]	15
Rys. 1.6 Robot Armeo Power (Hocoma) [22].....	16
Rys. 1.7 Robot ReoGo (Motorika) [23]	17
Rys. 1.8 Robot HARMONY [24].....	17
Rys. 1.9 Robot DIEGO (TyroMotion) [25].....	18
Rys. 1.10 Robot LUNA (EgzoTech) [26]	18
Rys. 1.11 Robot ARM-100 (ITAM Zabrze) [28].....	19
Rys. 3.1 Schemat struktury pracy.....	Błąd! Nie zdefiniowano zakładki.
Rys. 4.1 Zdjęcie modułu Slave robota ARM-100 wraz z napędem i czujnikami.....	Błąd! Nie zdefiniowano zakładki.
Rys. 4.2 Zdjęcie robota ARM-100	Błąd! Nie zdefiniowano zakładki.
Rys. 4.3 Wizualizacja robota ARM-100 za pomocą biblioteki OpenGL.....	Błąd! Nie zdefiniowano zakładki.
Rys. 4.4 Sposób rozmieszczenia markerów pasywnych na ramieniu prototypu urządzenia rehabilitacyjnego [37]	Błąd! Nie zdefiniowano zakładki.
Rys. 5.1 Zdjęcie robota ARM-100	28
Rys. 5.2 Wizualizacja par kinematycznych robota ARM-200 z zaznaczonymi stopniami swobody (1-7 podstawowe stopnie swobody, 1'-3' pomocnicze stopnie swobody).....	30
Rys. 5.3 Schemat systemu sterowania robota ARM-200.....	31
Rys. 5.4 Schemat blokowy modułu Master wraz z zasilaniem i modułem przekaźników bezpieczeństwa.....	32
Rys. 5.5 Uproszczony schemat Modułu Przekazników Bezpieczeństwa i elementów towarzyszących.....	33
Rys. 5.6 Schemat blokowy modułu Slave systemu sterowania robotem.....	35

Rys. 5.7 Schemat połączenia w parę dwóch modułów Slave	36
Rys. 5.8 Schemat pracy modułu Hamulca Dynamicznego	37
Rys. 5.9 Schemat blokowy modułu sEMG	38
Rys. 5.10 Zdjęcie modułu sEMG wraz z przewodami.....	39
Rys. 5.11 Zdjęcie przedstawiające czujnik siły w uchwycie przedramienia robota ARM-200.....	39
Rys. 5.12 Rysunek czujnika siły EMS30 wraz ze schematem połączeń [54]	40
Rys. 5.13 Zdjęcia bezwzględno enkodera pozycji kątowej.....	40
Rys. 5.14 Zdjęcie zabudowanego czujnika bezwzględnej pozycji kątowej węzle nr 1 robota ARM-200	41
Rys. 5.15 Wizualizacja umieszczenia magnesu trwałego względem elementu pomiarowego AS5600.....	41
Rys. 5.16 Schematyczny opis magistrali robota, składającej się z magistrali danych RS-485, linii alarmowej i zasilania	43
Rys. 5.17 Sygnały logiczne magistrali RS-485 podczas wymiany danych między modułem Master i modułami Slave.....	43
Rys. 5.18 Schemat komunikacji w systemie sterowania robota	44
Rys. 5.19 Uproszczony schemat połączenia modułu Master z modułami Slave	45
Rys. 5.20 Zdjęcie modułów Slave robota ARM-200 połączonych magistralą	45
Rys. 5.21 Schemat działania linii alarmowo-adresowej	46
Rys. 5.22 Sygnały automatycznego adresowania modułów Slave	47
Rys. 5.23 Ekran inżynierskiej aplikacji sterującej robotem „ARM-200 DataBusTool”	48
Rys. 5.24 Okno programu VAST.Rehab współpracujące z robotem ARM-200	49
Rys. 5.25 Fragment okna aplikacji odpowiedzialny za nawiązanie połączenia z robotem	50
Rys. 5.26 Kontrolki połączenia z węzłami robota i przyciski globalnego sterowania.....	50
Rys. 5.27 Fragment okna odpowiedzialny za rysowanie wykresów wybranych parametrów	50
Rys. 5.28 Fragment okna odpowiedzialny głównie za wyświetlanie wartości liczbowych wybranych parametrów	51
Rys. 5.29 Fragment okna z suwakami umożliwiającymi zmianę podstawowych parametrów pracy węzła.....	51

Rys. 5.30 Fragment okna, w którym jest możliwość zmiany parametrów regulatorów PID.....	52
Rys. 5.31 Listwa dolna okna programu wyświetlająca informacje statusowe.....	52
Rys. 5.32 Uproszczony schemat działania programu głównego modułu Master	53
Rys. 5.33 Uproszczony schemat działania programu odbioru danych modułu Master	54
Rys. 5.34 Uproszczony schemat działania programu głównego modułu Slave.....	55
Rys. 5.35 Schemat działania pojedynczego węzła robota.....	58
Rys. 5.36 Opis algorytmu sterowania w trybie śledzenia ruchu	59
Rys. 5.37 Schemat blokowy sterowania w trybie śledzenia ruchu	60
Rys. 5.38 Opis algorytmu sterowania w trybie odtwarzania ruchu	61
Rys. 5.39 Schemat blokowy sterowania w trybie odtwarzania ruchu.....	62
Rys. 5.40 Uogólniony schemat pracy regulatora	63
Rys. 5.41 Budowa regulatora PID: a - klasyczna kaskadowa, b- równoległa.....	64
Rys. 5.42 Odpowiedź obiektu na skok jednostkowy	65
Rys. 5.43 Symulacja regulatora PI dla parametrów wyliczonych metodą Ziglera-Nicholsa [61]	68
Rys. 5.44 Symulacja regulatora PI dla parametrów wyliczonych metodą Lambda [61]	69
Rys. 5.45 Symulacja regulatora PI dla parametrów wyliczonych metodą Lambda z modyfikacją [61]	70
Rys. 5.46 Opis sekwencji komutacji silnika BLDC z czujnikami Halla [64, s. 67]	71
Rys. 5.47 Schemat sposobu sterowania fazami silnika BLDC	72
Rys. 5.48 Komutacja faz silnika BLDC z modulacją PWM.....	73
Rys. 5.49 Uproszczony schemat enkapsulacji przesyłanych danych.....	74
Rys. 5.50 Przykład przesyłanego pakietu UDP w oknie programu Wireshark.....	77
Rys. 5.51 Przykład wizualizacji danych z wybranego węzła systemu sterowania robota rehabilitacyjnego	97
Rys. 6.1 Uproszczony schemat blokowy systemu sterowania robota rehabilitacyjnego	99
Rys. 6.2 Przedstawienie sposobu pomiaru siły nacisku	100
Rys. 6.3 Sposób sprawdzenia dokładności pomiaru prędkości obrotowej.....	102
Rys. 6.4 Zdjęcia magnetycznego czujnika bezwzględnej pozycji kątowej.....	103
Rys. 6.5 Schemat badania dokładności toru pomiarowego prądu.....	105
Rys. 6.6 Widok zmienności czasu powtarzania Tp.....	109

Rys. 6.7 Rysunek prezentuje hipotetyczny efekt nieregularnej komunikacji	109
Rys. 6.8 Schemat przepływu informacji podczas eksperymentu	110
Rys. 6.9 Syntetyczna trajektoria ruchu ramienia robota (zapętlona w aplikacji sterującej)	110
Rys. 6.10 Wykresy czasu pomiędzy próbkami i pozycji kątowej ramienia robota dla klasy Timers	111
Rys. 6.11 Wykresy czasu pomiędzy próbkami i pozycji kątowej ramienia robota dla klasy Stopwatch.....	112
Rys. 6.12 Pomiar czasu wymaganego na obsługę pełnego cyklu wymiany danych.....	113
Rys. 6.13 Reakcja systemu sterowania na brak odpowiedzi modułu Slave nr2	114
Rys. 6.14 Sekwencja ruchów podczas rotacji w stawie łokciowym, węzeł 7 Błąd! zdefiniowano zakładki.	Nie
Rys. 6.15 Sekwencja ruchów podczas zgięcia dłoniowo-grzbietowego, węzeł 6 Błąd! zdefiniowano zakładki.	Nie
Rys. 6.16 Wykresy parametrów węzła 7 dla rotacji w stawie łokciowym Błąd! zdefiniowano zakładki.	Nie
Rys. 6.17 Wykresy parametrów węzła 6 podczas zgięcia dłoniowo-grzbietowego Błąd! zdefiniowano zakładki.	Nie
Rys. 6.18 Zdjęcie robota ARM-200 w konfiguracji na kończynę prawą.....	115
Rys. 6.19 Zmiany pozycji w 7 węzłach robota rehabilitacyjnego ARM-200	116
Rys. 6.20 Węzeł 1 - odwiedzenie w stawie barkowym.....	116
Rys. 6.21 Węzeł 2 - zgięcie w stawie barkowym.....	117
Rys. 6.22 Węzeł 3 - rotacja w stawie barkowym	117
Rys. 6.23 Węzeł 4 - zgięcie w stawie łokciowym.....	117
Rys. 6.24 Węzeł 5 - zgięcie promieniowo-łokciowe	118
Rys. 6.25 Węzeł 6 - zgięcie dłoniowo-grzbietowe.....	118
Rys. 6.26 Węzeł 7 - rotacja w stawie łokciowym	118
Rys. 6.27 Ustawienie stawu łokciowego: a) zgięcie 90°, b) wyprost 0°	119
Rys. 6.28 Badanie węzła nr 4 robota ARM-200 z wyłączonymi progami nieczułości siły	119
Rys. 6.29 Badanie węzła nr 4 robota ARM-200 z ustawionym progami nieczułości siły na 3 N	120

Rys. 6.30 Badanie węzła nr 4 robota ARM-200 w odwiedzeniu kończyny	121
Rys. 6.31 Badanie prędkości ruchu ramienia robota ARM-200 w węźle nr 4.....	121
Rys. 6.32 Pomiar prędkości od przyłożonej siły wraz z oczekiwaną odpowiedzią układu (kolor czerwony)	122
Rys. 6.33 Ustawienie ramienia robota ARM-200 podczas badania.....	123
Rys. 6.34 Odpowiedzi czujników sił robota ARM-200, przy wywieraniu siły na czujnik nr 7.....	123
Rys. 6.35 Wykres wartości sił w węzłach robota, podczas zginanie w stawie łokciowym	125
Rys. 6.36 Wykres modułów wartości sił w węzłach robota, podczas zginanie w stawie łokciowym	126

SPIS TABEL

Tab. 4.1 Wbudowane środki bezpieczeństwa w odniesieniu do PN-EN ISO 13482	Błąd! Nie zdefiniowano zakładki.
Tab. 4.2 Lista środków ochronnych	Błąd! Nie zdefiniowano zakładki.
Tab. 5.1 Tabela możliwych stanów modułu Master	54
Tab. 5.2 Tabela możliwych trybów pracy modułu Slave	57
Tab. 5.3 Wyliczenia współczynników regulatora wg metody Ziglera-Nicholsa	66
Tab. 5.4 Wyliczenia współczynników regulatora wg metody Lambda	67
Tab. 5.5 Szczegółowy opis warstw i danych pakietu	75
Tab. 5.6 Ogólny opis struktury danych pakietu UDP robota ARM-200	77
Tab. 5.7 Ramka rozkazów z PC do modułu Slave	78
Tab. 5.8 Ramka odpowiedzi z Slave do PC przesyłane są w postaci ramek danych	79
Tab. 5.9 Lista typów danych data_c_type rozkazów przesyłanych z PC do modułów	80
Tab. 5.10 Lista typów danych data_r_type odpowiedzi przesyłanych z modułów do PC	81
Tab. 5.11 Opis pakietu Control (0xA1)	81
Tab. 5.12 Opis pakietu Config1 (0xA2)	82
Tab. 5.13 Opis pakietu Config2 (0xA3)	83
Tab. 5.14 Opis pakietu Config3 (0xA3)	84
Tab. 5.15 Opis pakietu FirmwareVersion (0xA5)	86
Tab. 5.16 Tryby pracy modułu Slave	86
Tab. 5.17 Tryby pomiaru siły modułu Slave	87
Tab. 5.18 Strona pracy modułu Slave	88
Tab. 5.19 Bity statusu NodeWarnings modułu Slave	88
Tab. 5.20 Opis pakietu ControlReplay (0xB1)	89
Tab. 5.21 Opis pakietu Config1Replay (0xB2)	90
Tab. 5.22 Opis pakietu Config2Replay (0xB3)	91
Tab. 5.23 Opis pakietu Config3Replay (0xB4)	92
Tab. 5.24 Opis pakietu FirmwareVersionReplay (0xB5)	94
Tab. 5.25 Opis pakietu ControlMasterReplay (0xC1)	95
Tab. 6.1 Wyniki pomiaru dokładności toru pomiarowego siły	101

Tab. 6.2 Wyniki dokładności toru pomiaru prędkości obrotowej.....	102
Tab. 6.3 Wyniki pomiarów dokładności bezwzględnego czujnika pozycji kątowej	104
Tab. 6.4 Wyniki badania dokładności pomiaru prądu silnika BLDC	105
Tab. 6.5 Wyniki badania dokładności pomiaru napięcia zasilania	107
Tab. 6.6 Wyniki badania nieintencjonalnej aktywności czujników sił robota ARM-200 .	124
Tab. 6.7 Wartości sił i prędkości zarejestrowanych podczas badania z progiem nieczułości 6,4 N	126