



Praca dyplomowa magisterska

Adam Skubis

**Modelowanie manipulatora IRp-6 ze  
sterownikiem siłowym  
wykorzystującym odczyty z jednostki  
inercyjnej**

Praca dyplomowa magisterska pod kierunkiem  
dr inż. Tomasza Winiarskiego

Instytut Automatyki i Informatyki Stosowanej  
Politechniki Warszawskiej

Warszawa, Wrzesień 2013

## **Podziękowania**

Chciałbym podziękować wszystkim osobom, które pomogły mi w stworzeniu tej pracy. W szczególności podziękowania należą się mojemu promotorowi, Tomaszowi Winiarskiemu za wszystkie uwagi i propozycje dotyczące tej pracy, jakie od niego otrzymałem, a także za ciągłą pomoc w kolejnych etapach pracy. Dziękuję również innym osobom, które pomagały mi podczas pisania tej pracy - Konradowi Banachowiczowi, Magdalenie Kuli, Maciejowi Stefańczykowi, Bartoszowi Świstakowi i Michałowi Walęckiemu.

## Streszczenie

Celem niniejszej pracy magisterskiej było określenie parametrów układu zawierającego regulator siłowy działający na manipulatorze IRp-6. Regulator ten miał wykorzystywać pomiary siły bezwładności robota uzyskane z jednostki inercyjnej. W tym celu przeprowadzone zostały eksperymenty identyfikujące wybrane parametry manipulatora. Na podstawie tych parametrów zostały wyznaczone parametry opisujące transmitancję kolejnych członów modelu manipulatora. Identyfikacja została przeprowadzona zarówno wykorzystywanymi już w laboratorium metodami jak i nowymi, automatycznymi metodami wyznaczania parametrów transmitancji na podstawie sygnałów wejściowych i wyjściowych.

**Słowa kluczowe:** *regulator, robot IRp-6, dobór parametrów, jednostka inercyjna*

## Abstract

**Title:** **Modelling of an IRp-6 manipulator with force controller using inertive unit readings.**

The aim of this thesis was to determine parameters of a structure containing force controller and IRp-6 manipulator. This controller was to use inertia readings from an inertial measurement unit. To accomplish this a series of experiments was conducted to identify some system parameters. Based on these parameters other parameters, defining transmittance of manipulators model segments were computed. The identification was conducted using already tested methods as well as new, automatic methods of defining transmittance parameters based on input and output signals.

**Słowa kluczowe:** *controller, IRp-6 robot, parameter adjustment, inertial measurement unit*



# Spis treści

<b>1</b>	<b>Wstęp</b>	<b>7</b>
1.1	Geneza pracy . . . . .	7
1.2	Cel pracy . . . . .	7
1.3	Układ pracy . . . . .	8
<b>2</b>	<b>Pośrednie sterowanie siłowe</b>	<b>9</b>
2.1	Pośrednie sterowanie siłowe . . . . .	9
2.2	Wykorzystanie jednostki inercyjnej . . . . .	10
2.3	Model systemu robotycznego . . . . .	10
<b>3</b>	<b>Sprzęt i oprogramowanie</b>	<b>15</b>
3.1	Manipulator IRp-6 . . . . .	15
3.2	Struktura ramowa MRROC++ . . . . .	17
3.2.1	Struktura MRROC++ . . . . .	18
3.2.2	Generator TFF_NOSE_RUN . . . . .	19
3.3	MATLAB . . . . .	20
3.3.1	SIMULINK . . . . .	20
3.3.2	System Identification Tool . . . . .	21
3.4	LabVIEW . . . . .	21
<b>4</b>	<b>Stworzone oprogramowanie</b>	<b>23</b>
4.1	Zadanie pierwsze . . . . .	23
4.2	Zadanie drugie . . . . .	25
4.2.1	Zadanie w MRROC++ . . . . .	25
4.2.2	Generator SINUSOIDAL VELOCITY . . . . .	27
4.3	Programy symulacyjno-obliczeniowe w środowisku MATLAB . . . . .	30
4.4	Program obliczeniowy w LabVIEW . . . . .	32
<b>5</b>	<b>Identyfikacja parametrów układu</b>	<b>35</b>
5.1	Budowa stanowiska badawczego . . . . .	36

5.2	Eksperyment 1. . . . .	36
5.2.1	Sposób przeprowadzenia eksperymentu . . . . .	37
5.2.2	Wyniki eksperymentu . . . . .	37
5.3	Eksperyment 2. . . . .	41
5.3.1	Sposób przeprowadzenia eksperymentu . . . . .	41
5.3.2	Wyniki eksperymentu . . . . .	41
5.4	Eksperyment 3. . . . .	45
5.4.1	Sposób przeprowadzenia eksperymentu . . . . .	45
5.4.2	Wyniki eksperymentu . . . . .	45
5.5	Eksperyment 4. . . . .	48
5.5.1	Sposób przeprowadzenia eksperymentu . . . . .	48
5.5.2	Wyniki eksperymentu . . . . .	48
5.6	Eksperyment 5. . . . .	50
5.6.1	Sposób przeprowadzenia eksperymentu . . . . .	50
5.6.2	Wyniki eksperymentu . . . . .	51
5.7	Eksperyment 6. . . . .	53
5.7.1	Sposób przeprowadzenia eksperymentu . . . . .	53
5.7.2	Wyniki eksperymentu . . . . .	53
5.8	Podsumowanie eksperymentów . . . . .	55
<b>6</b>	<b>Weryfikacja pomiarów z użyciem jednostki inercyjnej</b>	<b>57</b>
6.1	Badanie opóźnienia . . . . .	57
6.2	Badanie wpływu opóźnienia na poprawę odczytów siły . . . . .	58
<b>7</b>	<b>Identyfikacja członu <math>G_2</math></b>	<b>61</b>
7.1	Cel badań . . . . .	61
7.2	Sposób przeprowadzenia badań . . . . .	62
7.3	Badanie 1. . . . .	64
7.4	Badanie 2. . . . .	64
7.5	Badanie 3. . . . .	65
7.6	Badanie 4. . . . .	66
7.7	Weryfikacja działania zidentyfikowanego członu $G_2$ w układzie . . . .	66
7.8	Wnioski z badań . . . . .	66
<b>8</b>	<b>Wnioski</b>	<b>69</b>
	<b>Bibliografia</b>	<b>70</b>

# Rozdział 1

## Wstęp

### 1.1 Geneza pracy

W laboratorium robotyki Instytutu Automatyki i Informatyki Stosowanej Politechniki Warszawskiej zamontowane zostały na manipulatorach IRp-6 jednostki inercyjne [1]. Jednostki te zostały zamontowane na nadgarstkach manipulatorów. Celem użycia tych elementów jest poprawa sterowania robotami na podstawie oceny siły bezwładności manipulatora. Użycie jednostki inercyjnej zmieniło parametry układu robotycznego, w związku z czym należy je ponownie zidentyfikować. Dodatkowo uznano, że inne metody niż dotychczasowe mogą pozwolić na weryfikację otrzymanych parametrów.

### 1.2 Cel pracy

Celem tej pracy jest identyfikacja parametrów modelu robotycznego manipulatora IRp-6 ze sterownikiem siłowym korzystającym z jednostki inercyjnej. W trakcie pracy przeprowadzone zostaną eksperymenty na rzeczywistym obiekcie, oraz komputerowe symulacje modelu robota. Celem tych symulacji będzie dobranie i weryfikacja parametrów członów transmitancji systemu robotycznego poprzez porównanie wyników tych symulacji z wynikami eksperymentów. Wykorzystane zostaną także automatyczne metody pozwalające na identyfikację parametrów funkcji transmitancji o zadanej postaci na podstawie sygnałów wejściowych i wyjściowych. Następnie przeanalizowane zostaną niedokładności wyników i określone zostaną ich przyczyny.

## 1.3 Układ pracy

W rozdziale 2. opisane zostały teoretyczne podstawy do stworzenia tej pracy. Omówione jest zagadnienie tworzenia regulatora dla systemu oraz opisany jest wykorzystywany w tej pracy model układu robotycznego. Rozdział 3. zawiera opis technologii użytych w tej pracy - manipulatora IRp-6 oraz używanej w celu sterowania nim struktury ramowej MRROC++. W tym rozdziale omówiono także programy użyte do analizy danych pobranych w eksperymentach i symulacji - MATLAB z pakietem SIMULINK i narzędziem System Identification Toolbox oraz LabVIEW. W rozdziale 4. znajduje się opis stworzonych programów w środowisku MATLAB, LabVIEW oraz zadań robotycznych w MRROC++. Rozdział 5. zawiera opis stanowiska, na którym przeprowadzone zostały eksperymenty oraz przedstawia eksperymenty użyte do identyfikacji parametrów układu. Opisany został sposób przeprowadzenia eksperymentów, a także przedstawione zostały w formie wykresów ich wyniki. Następnie opisane zostały działania prowadzące do uzyskania parametrów układu robotycznego. W rozdziale 6. znajduje się opis przeprowadzonych działań w celu weryfikacji poprawności pomiarów wykonanych z wykorzystaniem pomiarów z jednostki inercyjnej. Rozdział 7. zawiera opis badań przeprowadzonych w celu identyfikacji transmitancji członu  $G_2$ . Rozdział 8. zawiera podsumowanie pracy, oraz płynące z niej wnioski.



# Rozdział 2

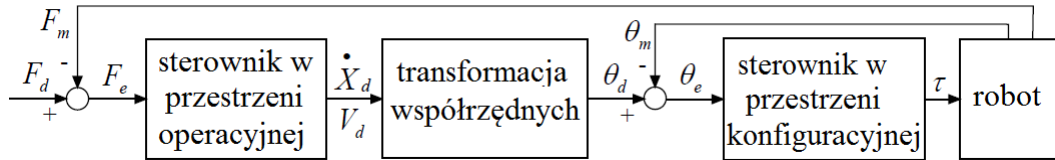
## Pośrednie sterowanie siłowe

W tym rozdziale przedstawiona została teoria wykorzystywana w tej pracy. W rozdziale 2.1. znajduje się opis pośredniego sterowania siłowego. Rozdział 2.2. opisuje sterowanie siłowe robotem z wykorzystaniem jednostki inercyjnej. Rozdział 2.3. zawiera opis systemu robotycznego, parametrów je opisujących, oraz schemat tego układu w pakiecie SIMULINK.

### 2.1 Pośrednie sterowanie siłowe

W tej pracy skupiać się będę na pośrednim sterowaniu siłowym [2]. Algorytm ten składa się z dwóch pętli sterowania:

- Zewnętrznej, sterowanej siłowo,
- Wewnętrznej, sterowanej pozycyjnie.



Rysunek 2.1: Schemat pośredniego sterowania siłowego

W pośrednim sterowaniu siłowym sterownik zadaje prędkość  $\dot{X}$  przedstawioną w przestrzeni operacyjnej. Ta prędkość jest przetwarzana na zadaną pozycję w przestrzeni konfiguracyjnej  $\theta$  przez człon transformacji współrzędnych. Kontroler przestrzeni konfiguracyjnej nakazuje robotowi (jego silnikom) dotarcie do pozycji  $\theta$ . Schemat takiego sposobu regulacji znajduje się na rysunku 2.1.

## 2.2 Wykorzystanie jednostki inercyjnej

W sterowaniu siłowym istotnymi danymi są odczyty siły z czujnika. Należy jednak zauważyć, że poza siłą kontaktu, która powinna być mierzona, czujnik siły robota mierzy także inercję. Można ją rozdzielić na następujące składowe:

- Siłę grawitacji,
- Siłę odśrodkową,
- Siłę Coriolisa,
- Siłę bezwładności.

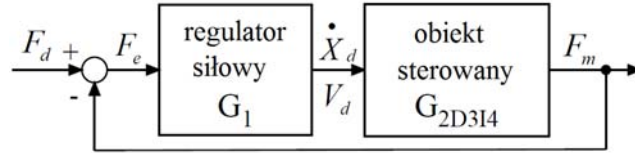
Wykrycie aktualnego ruchu końcówki robota byłoby wystarczające do kompensacji wpływu tych sił. W teorii wystarcza do tego odczytanie położenia silników, jednak z powodu opóźnienia transportowego pomiędzy położeniem silników a końcówki nie można tego sposobu zastosować z powodzeniem w większości aplikacji. W związku z tym w celu kompensacji sił stosowane są jednostki inercyjne [1, 3]. Zainstalowana w laboratorium robotyki jednostka inercyjna to moduł ADIS16362 firmy Analog Device zawierający trzy akcelerometry i trzy żyroskopy. Dane z tej jednostki przesyłane są do komputera sterującego co 2ms (krok sterowania) przy użyciu magistrali USB. W celu przekształcenia sił i momentów sił mierzonych podczas wykonywania ruchu niezbędna jest znajomość następujących parametrów:

- Masy chwytaka  $w$ ,
- Położenie środka ciężkości chwytaka  ${}^W_G P$ , gdzie  $G$  jest układem związanym ze środkiem ciężkości chwytaka, a  $W$  układem związanym z nadgarstkiem,
- Macierz  $6 \times 6$   $F$  transformująca siły działające na środek ciężkości chwytaka do układu związanego z nadgarstkiem.

W celu kompensacji inercji przyspieszenia uzyskane w jednostce inercyjnej transformowane są do układu współrzędnych nadgarstka i odejmowane od sił zmierzonych przez czujnik sił.

## 2.3 Model systemu robotycznego

Model systemu robotycznego robota IRp-6 w laboratorium robotyki można przedstawić jako regulator i obiekt sterowany jak na rysunku 2.2.

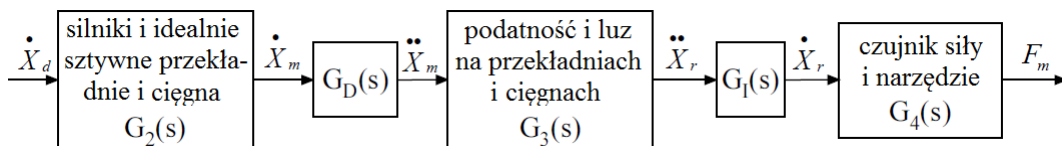


Rysunek 2.2: Ogólny model systemu robotycznego

Przedstawienie obiektu sterowanego (modelu robota IRp-6) znajduje się na rysunku 2.3.

Oznaczenia na rysunkach 2.2, 2.2. i 2.3. należy rozumieć jako:

- $\dot{X}_d, V_d$  - prędkość zadana - generowana przez sterownik,
- $\dot{X}_r$  - prędkość rzeczywista - uzyskana poprzez scałkowanie  $\ddot{X}_r$ ,
- $\dot{X}_m$  - zmierzona prędkość  $\dot{X}_m = \frac{(X_m^t - X_m^{t-1})}{\Delta t}$ ,
- $\ddot{X}_r$  - przyspieszenie rzeczywiste - oszacowane na podstawie symulacji,
- $\ddot{X}_m$  - zmierzone przyspieszenie  $\ddot{X}_m = \frac{(\dot{X}_m^t - \dot{X}_m^{t-1})}{\Delta t}$ ,
- $F_m$  - siła zmierzona przez czujnik siły,
- $F_e$  - błąd siły, uzyskany poprzez odjęcie od siły zadanej siły zmierzonej,
- $F_d$  - siła zadana,
- $\theta_m$  - pozycja zmierzona w przestrzeni konfiguracyjnej,
- $\theta_d$  - pozycja zadana w przestrzeni konfiguracyjnej,
- $\theta_e$  - błąd pozycji w przestrzeni konfiguracyjnej,
- $\tau$  - sterowanie wysyłane do robota.



Rysunek 2.3: Przybliżony model robota IRp-6

Kolejnymi elementami układu sterowania są:

- Regulator siłowy  $G_1$ ,
- Silniki i idealnie sztywne przekładnie i cięgna  $G_2$ ,

- Różniczkowanie, reprezentowane przez funkcję ( $G_D(s) = s$ ),
- Podatność i luz na przekładniach i cięgnach  $G_3$ ,
- Całkowanie, reprezentowane przez funkcję ( $G_I(s) = \frac{1}{s}$ ),
- Czujnik siły i narzędzie  $G_4$ .

Transmitancja systemu z pętlą zamkniętą prezentowana jest przez wzór:

$$G_Z(s) = \frac{G_{12D3I4}(s)}{1 + G_{12D3I4}(s)} \quad (2.1)$$

Transmitancje powyższych elementów można zapisać w przybliżeniu wzorami:

$$G_1 = \frac{\frac{1}{Bp}}{\frac{Mp}{Bp} * s + 1} \quad (2.2)$$

$$G_2 = \frac{1}{(Ta * s + 1) * (Tb * s + 1)} \quad (2.3)$$

$$G_D(s) = s \quad (2.4)$$

$$G_3 = e^{(-s * Td)} \quad (2.5)$$

$$G_I(s) = \frac{1}{s} \quad (2.6)$$

$$G_4 = M * s + B + \frac{K}{s} \quad (2.7)$$

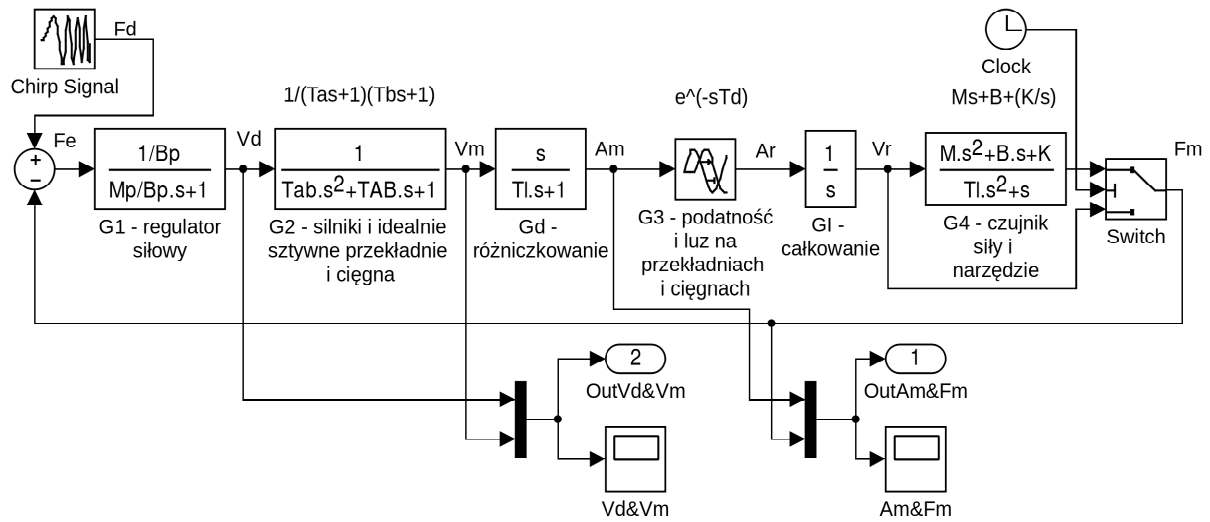
Z powyższych równań wynika, że aby poprawnie określić model należy znaleźć wartości poniższych parametrów systemu robotycznego:

- $T_a, T_b$  - człon inercyjny 2. rzędu, dobrany aby zapewnić zależność pomiędzy mierzoną prędkością  $\dot{X}_m$  i zadaną prędkością  $\dot{X}_d$ ,
- $T_d$  - opóźnienie czasowe pomiędzy przyspieszeniem  $\ddot{X}_m$  a  $\ddot{X}_r$ .  $\ddot{X}_r$  można zmierzyć w przypadku braku kontaktu robota z otoczeniem - gdy jedyną siłą jest siła inercji,
- $M$  - masa narzędzia,
- $B$  - tłumienie (występuje, gdy istnieje kontakt końcówki robota z otoczeniem),
- $K$  - sztywność (uznana za zero, wszelkie właściwości kontaktu przypisywane są tłumieniu).

Dodatkowymi parametrami tego układu są znane wartości tłumienia i inercji regulatora:

- $B_p$  - tłumienie regulatora, które jest ustawiane w algorytmie sterowania (w związku z tym jest dokładnie znane),
- $M_p$  - inercja regulatora, podobnie jak tłumienie jest ustawiana w algorytmie i dokładnie znana.

Model robota IRp-6 w środowisku SIMULINK odzwierciedlający powyższą budowę układu i wartości transmitancji znajduje się na rysunku 2.4



Rysunek 2.4: Model manipulatora IRp-6 w środowisku SIMULINK

Z powodu nierealizowalności fizycznej transmitancji o wyższym stopniu potęgi  $s$  w liczniku niż w mianowniku (o większej liczbie zer niż biegunów) transmitancje w tej postaci są rozszerzane w mianowniku o najwyższy stopień potęgi  $s$  obecny w liczniku pomnożony przez bardzo małą stałą  $Tl$ .



# Rozdział 3

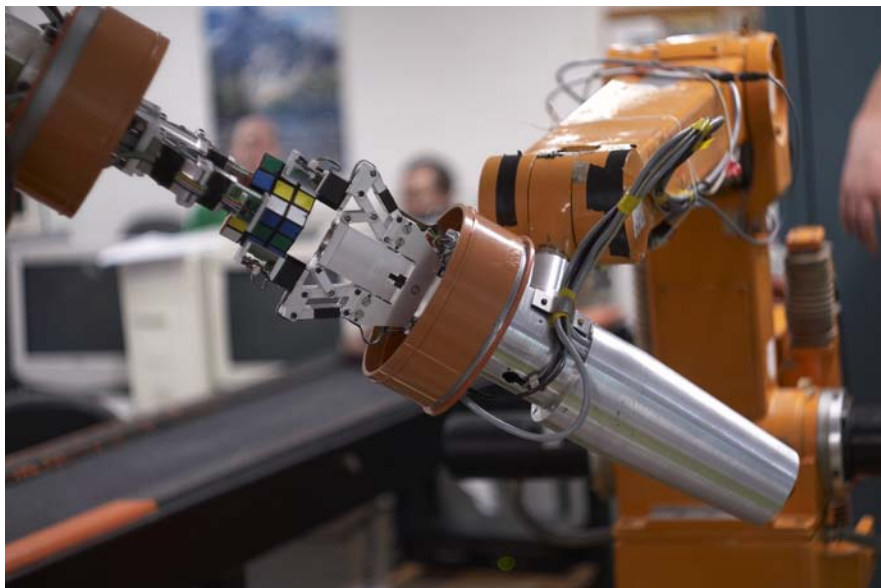
## Sprzęt i oprogramowanie

W tym rozdziale przedstawione zostały sprzęt i oprogramowanie wykorzystane przy tworzeniu tej pracy. W rozdziale 3.1. znajduje się opis robotów IRp-6, scharakteryzowane są też układy pomiarowe i sterownicze tego robota. Rozdział 3.2. zawiera opis struktury MRROC++, budowy zadań w tej strukturze, roli, jaką pełnią poszczególne procesy, oraz wykorzystywanego regulatora. Rozdział 3.3. zawiera opis pakietu MATLAB, oraz używanych dodatków - pakietu SIMULINK oraz System Identification Tool. W rozdziale 3.4. opisane zostało środowisko LabVIEW, podstawowe informacje o programach w tym środowisku oraz najważniejsze używane funkcje biblioteczne.

### 3.1 Manipulator IRp-6

Manipulator jest to robot, mający na celu pełnienie roli ludzkich kończyn górnych - dosięgnięcie do obiektu i manipulację nim. Każdy manipulator ma określone przestrzenie: roboczą i manipulacyjną w których może wykonywać zadania. Przestrzenie te są ściśle związane z kinematyką danego manipulatora.

Manipulator IRp-6 [4] zbudowany został w Polsce w Przemysłowym Instytucie Automatyki i Pomiarów (PIAP). Jest to manipulator stawowy, który powstał na licencji szwedzkiej firmy ASEA (Allemana Svenska Elektriska Aktienbolaget). Został stworzony jako manipulator przemysłowy, wykorzystywany najczęściej do lakierowania i spawania części. Robot IRp-6 może być sterowany zarówno ręcznie przez operatora jak i wykonywać zaprogramowane zadania. Przy użyciu systemu wizyjnego może wykonywać czynności oparte na rozpoznawaniu obrazów. Operator manipulatora nie musi być w bezpośrednim otoczeniu części manipulacyjnej, żeby wykonać zadanie, gdyż może używać systemu robotycznego do zaprogramowania manipulatora. Należy wyróżnić dwa rodzaje funkcji działania manipulatora: manipulacyjną,



Rysunek 3.1: Manipulatory IRp-6 układające kostkę Rubika

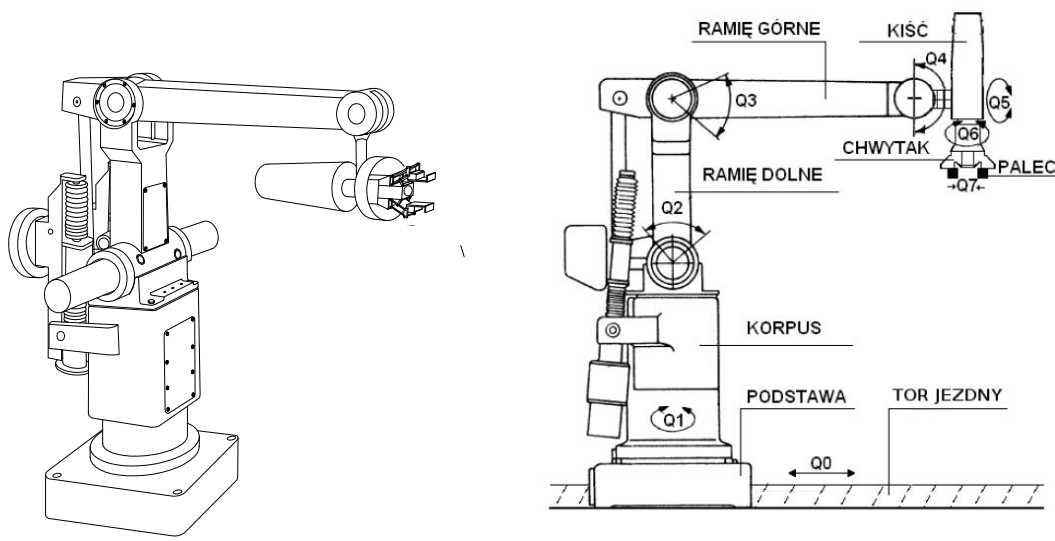
wykonywaną przez chwytak i wysięgnikową, realizowaną przez ramię manipulatora.

Rysunek 3.1. przedstawia manipulatory IRp-6 znajdujące się w laboratorium robotyki Instytutu Automatyki i Informatyki Stosowanej Politechniki Warszawskiej. IRP-6 jest przykładem robota przemysłowego i najczęściej używany jest do automatyzacji procesów przemysłowych. Wykorzystuje się go głównie wtedy, kiedy życie lub zdrowie człowieka narażone byłoby na szkodę.

W laboratorium 012 Instytutu Automatyki i Informatyki stosowanej znajdują się dwa manipulatory IRp-6: *Track* i *Postument*. Obydwa wyposażone są w dwupalczaste chwytaki o jednym stopniu swobody. *Track* dodatkowo posiada jeszcze tor jezdny, po którym może się poruszać, natomiast *Postument*, jest na stałe zamontowany na podstawie. Na rysunku 3.2(a). przedstawiono przestrzenny schemat manipulatora o 7 stopniach swobody. Na rysunku 3.2(b). przedstawiono łańcuch kinematyczny manipulatora o 8 stopniach swobody:

- ruch po torze jezdny ( $Q_0$ ),
- obrót korpusu względem podstawy ( $Q_1$ ),
- obrót ramienia (ramienia dolnego) ( $Q_2$ ),
- obrót przedramienia (ramienia górnego) ( $Q_3$ ),
- pochylanie kiści (przegubu) ( $Q_4$ ),
- skręcanie kiści (przegubu) ( $Q_5$ ),
- obracanie chwytaka ( $Q_6$ ),
- ruch palcami ( $Q_7$ ).





(a) schemat przestrzenny robota IRp-6

(b) łańcuch kinematyczny robota IRp-6

Rysunek 3.2: Wykres siły zmierzonej i przyspieszenia zmierzonego

Na każdej osi stopnia swobody manipulatorów działają silniki elektryczne prądu stałego z przekładniami. Manipulator IRp-6 składa się z trzech zasadniczych układów: układu kinematycznego, układów pomiarowych i napędowych oraz z układu sterowania.

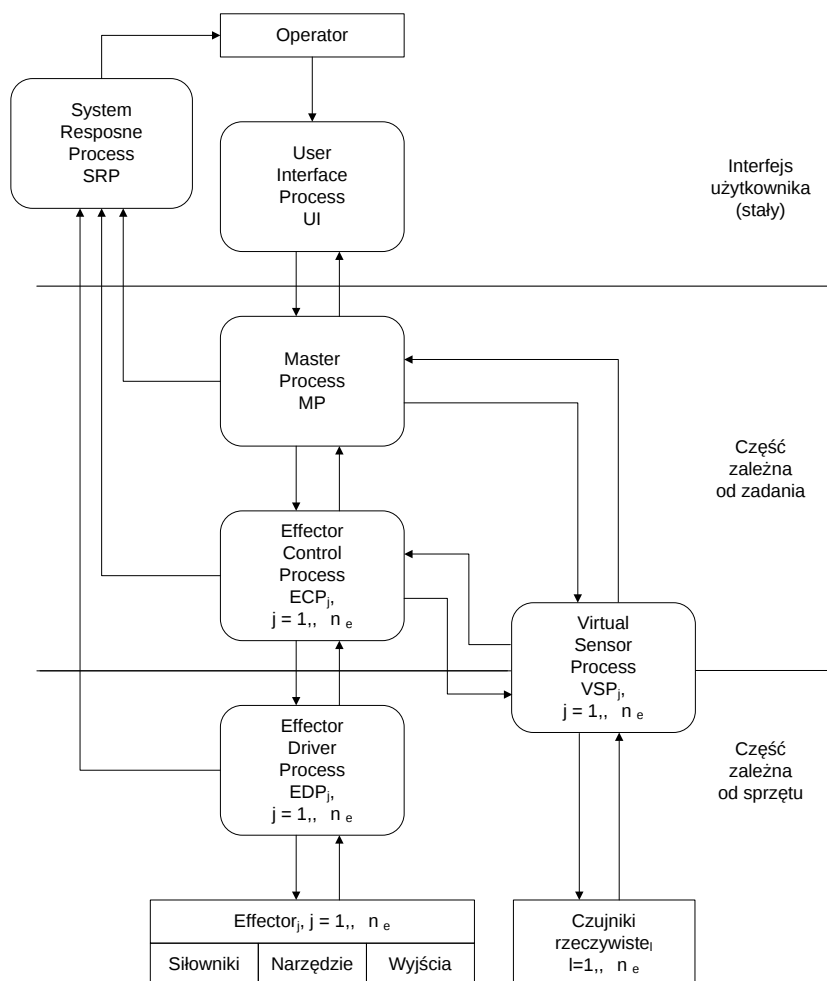
Układ kinematyczny odpowiada za przekształcanie działania obrotowego silników na ruch odpowiednich ramion manipulatora. W skład tego układu wchodzi wszystkie elementy kinematyczne robota wraz z elementami przeniesienia napędu: silniki, cięgna i przekładnie. Układy pomiarowe i napędowe odpowiedzialne są za regulację położenia robota. Na poszczególnych osiach działają sterowniki mocy oraz silniki prądu stałego. Układ sterowania natomiast składa się z komputera, pamięci, urządzeń wejścia wyjścia oraz modułów sterujących serwomechanizmami manipulatora.

## 3.2 Struktura ramowa MRROC++

Struktura ramowa MRROC++ (Multi-Robot Research Oriented Controller) [5] jest platformą zapewniającą biblioteki modułów oraz wzorce programistyczne do tworzenia sterowników dla systemów wielorobotowych. Pozwala ona na tworzenie zadań wielorobotowych w oparciu o istniejące już sterowniki i moduły.

### 3.2.1 Struktura MRROC++

System MRROC++ posiada hierarchiczną strukturę, dzięki czemu możliwe jest tworzenie sterowników składających się z wielu modułów. Każdemu modułowi mogą odpowiadać procesy, które uruchamiane są w systemie Linux. Dzięki takiej strukturze rozszerzanie o nowe zadanie nie wymaga pisania całości programu, jedynie potrzebnych (nowych) jego części. W strukturze MRROC++ występują następu-



Rysunek 3.3: Struktura systemu MRROC++

jące procesy:

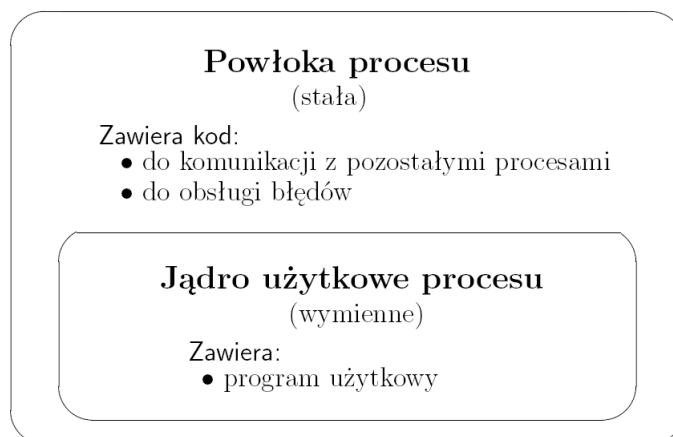
- **UI** (User Interface) - Interfejs, przez który użytkownik komunikuje się z programem,
- **MP** (Master Process) - Proces nadzorczy, kontrolujący pozostałe procesy,
- **ECP** (Effector Control Process) - Proces kontrolera efektora, zależny od zadania,

- **EDP** (Effector Driver Process) - Proces sterownika efektoru, zależny od użytego efektora,
- **VSP** (Virtual Sensor Process) - Proces czujnika wirtualnego, odpowiedzialny za odczyt i przetwarzanie danych z czujników rzeczywistych,
- **SRP** (System Response Process) - wątek odpowiedzi systemu (należący do procesu UI), przekazuje (wyświetla) użytkownikowi komunikaty o stanie effektorów i sterowników i/lub błędach w systemie.

Schemat połączeń tych procesów znajduje się na rysunku 3.3.

Procesy MP, ECP oraz VSP mają wspólną strukturę, przedstawioną na rysunku 3.4:

- Jądro procesu odpowiada za realizację zadania, jest zależne od zadania,
- Powłoka procesu odpowiada za komunikację z innymi procesami i obsługę błędów.



Rysunek 3.4: Struktura procesu w MRROC++

### 3.2.2 Generator TFF\_NOSE\_RUN

W strukturze MRROC++ generatory znajdują się na poziomie procesu ECP. Mają na celu wyliczenie kolejnych wartości parametrów sterujących (siły, pozycji, prędkości itp.) na podstawie dostępnych danych z czujników.

W przypadku generatora TFF\_NOSE\_RUN jego zadaniem jest zadanie parametrów regulatorowi znajdującemu się w procesie EDP. Regulator jest elementem, który na podstawie dostępnych danych ma za zadanie wygenerowanie sygnału sterującego. Wykorzystywany w tej pracy regulator posiada dwa parametry - tłumienie

oraz inercję, które wprowadza. Parametry tego regulatora definiowane są w programie. Regulator tego typu w porównaniu do regulatora PI, którego zazwyczaj używa się do stworzenia filtra dolnoprzepustowego ma tę zaletę, że nie sumuje błędu pomiaru siły. Używany jest do utrzymywania zerowych odczytów siły w jednej osi robota. Transmitancja takiego regulatora wynosi:

$$G_1 = \frac{\frac{1}{Bp}}{\frac{Mp}{Bp} * s + 1} \quad (3.1)$$

### 3.3 MATLAB

MATLAB [6, 7] jest środowiskiem do wykonywania obliczeń matematycznych, numerycznych, wizualizacyjnych i symulacyjnych. Nazwa programu MATLAB pochodzi od słów Matrix Laboratory. MATLAB jest również językiem programowania wysokiego poziomu, o składni zbliżonej do języka C. Obecnie (2013) posiada około 80 bibliotek funkcji i około 30 aplikacji zwanych toolboxami.

Podstawowym typem danych w programie MATLAB jest macierz. MATLAB akceptuje zarówno macierze o elementach rzeczywistych jak i zespolonych. Zmienne używane w środowisku MATLAB traktowane są jak macierze, a wektory i skalary są uważane za szczególne przypadki macierzy. Nie ma potrzeby deklarowania typu i wymiaru zmiennej, jest to automatycznie robione na podstawie danych przypisywanych do tej zmiennej. W przeciwieństwie do większości języków programowania indeksowanie tablic nie zaczyna się od 0, lecz od 1.

MATLAB posiada terminal, w którym wyświetla wyniki wykonania poleceń, informacje o błędach i ostrzeżeniach, a także pozwala na wprowadzanie komend. Istnieje też możliwość tworzenia własnych skryptów, które zawierają zbiory komend dla programu. W programie MATLAB można pisać własne funkcje i stosować struktury danych. Pozwala również na pisanie aplikacji zorientowanych obiektowo. MATLAB zawiera zestawy funkcji pozwalających na rysowanie dwu i trójwymiarowych wykresów oraz wizualizację wyników obliczeń w postaci rysunków statycznych i animacji.

Funkcje MATLAB [8] mogą być używane przy przetwarzaniu wyników eksperymentów. Umożliwia wizualizację danych, używanie funkcji statystycznych oraz posiada wbudowane funkcje aproksymujące.

#### 3.3.1 SIMULINK

SIMULINK jest środowiskiem diagramów blokowych zintegrowanym z pakietem MATLAB. Zawiera on edytor graficzny pozwalający na modelowanie i symulację

systemów dynamicznych.

W ramach tej pracy rozwinięty został model systemu robotycznego stworzony przez dra Winiarskiego. Na tym modelu przeprowadzane były symulacje zachowania systemu robotycznego ze sterownikiem. Dokładniejszy opis tych działań przedstawiony jest w rozdziale 5.

### 3.3.2 System Identification Tool

System Identification Tool jest narzędziem pakietu MATLAB, które pozwala na identyfikację parametrów dynamicznych systemów na podstawie zmierzonych danych wejściowych i wyjściowych. Istnieje możliwość użycia danych zarówno w domenie czasu jak i częstotliwości. Identyfikacja parametrów jest przeprowadzana dla zadanej postaci funkcji transmitancji - użytkownik podaje liczbę zer i biegunów układu. Dokładniejszy opis tych działań znajduje się w rozdziale 7.

## 3.4 LabVIEW

LabVIEW [9] jest środowiskiem programowania graficznego przewidzianym jako środowisko wirtualnej instrumentacji, ale pozwalającej też na tworzenie aplikacji bez użycia sprzętu. Programy w LabVIEW nazywane są VI (Virtual Instrument) i składają się z 3 głównych części:

- Ikony reprezentującej program,
- Panelu czołowego zawierającego dane wejściowe i wynikowe, pozwalającego na interakcję programu z użytkownikiem,
- Diagramu blokowego, na którym zgodnie z paradygmatami programowania przepływowego pisane jest ciało programu (wykonywane w nim operacje).

W niniejszej pracy LabVIEW zostało użyte ze względu na intuicyjną możliwość implementacji funkcji potrzebnych do analizy sygnałów zebranych podczas eksperymentów.

Funkcja Peak Detector pozwala na znajdowanie dolin i szczytów sygnału. Możliwe jest podawanie takich argumentów jak minimalna wysokość bezwzględna oraz minimalna szerokość szczytów/dolin. Funkcja Auto Power Spectrum dla podanego sygnału i okresu próbkowania zwraca wartości mocy dla częstotliwości sygnału.



# Rozdział 4

## Stworzone oprogramowanie

W tym rozdziale omówione zostały programy stworzone i wykorzystywane w ramach tej pracy magisterskiej. W rozdziale 4.1. znajduje się opis zadania w systemie MRROC++ użytego do wykonania eksperymentów identyfikacyjnych przy użyciu regulatora. Rozdział 4.2. opisuje zadanie identyfikacyjne wykorzystujące nowy, opisany w rozdziale 4.2.2. generator ruchu. W rozdziale 4.3. znajduje się opis stworzonych programów w środowisku MATLAB. Rozdział 4.4. opisuje program służący do obliczeń parametrów systemu robotycznego na podstawie eksperymentów i symulacji, napisany w środowisku LabVIEW.

### 4.1 Zadanie pierwsze

W pierwszych trzech eksperymentach identyfikacja parametrów robota była przeprowadzona poprzez wzbudzanie robota do rezonansu. Użyty w tym celu został generator opisany w rozdziale 3.2.2. Parametry regulatora zostały dobrane tak, aby robot się samoczynnie wzbudzał lub podtrzymywał wzbudzone drgania. W strukturze MRROC++ podczas przeprowadzania tych eksperymentów używany był standardowy program ECP korzystający z tego generatora. Proces MP zawierał trzy instrukcje dla robota - dojście do pozycji, w której przeprowadzane były eksperymenty, kalibrację czujnika siły (wyzerowanie odczytów w danym położeniu), oraz uruchomienie regulatora, tak aby robot utrzymywał pozycję w pięciu współrzędnych, zaś w szóstej dążył przy użyciu regulatora do utrzymywania zerowego odczytu siły.

Na wydruku 4.1. znajduje się funkcja `main_task_algorithm` dla tego zadania.

```
1 void askubis_demo::main_task_algorithm(void)
2 {
3   sr_ecp_msg->message("Ruch do pozycji FRONT");
4 }
```

```

5  set_next_ecp_state(ecp_mp::generator::ECP_GEN_SMOOTH_JOINT_FILE_FROM_MP ,
    1,
    "../src/application/askubis_demo_irp6/trajectory_postument_joint.trj",
    lib::irp6p_m::ROBOT_NAME);
6  wait_for_task_termination(false, lib::irp6p_m::ROBOT_NAME,
    lib::irp6p_m::ROBOT_NAME);
7
8  sr_ecp_msg->message("Zerowanie odczytów sił");
9
10 set_next_ecp_state(ecp_mp::generator::ECP_GEN_BIAS_EDP_FORCE ,
    0, "", lib::irp6p_m::ROBOT_NAME);
11 wait_for_task_termination(false, lib::irp6p_m::ROBOT_NAME,
    lib::irp6p_m::ROBOT_NAME);
12
13 sr_ecp_msg->message("Uruchomienie regulatora");
14
15 set_next_ecp_state(ecp_mp::generator::ECP_GEN_TFF_NOSE_RUN ,
    (int)
    ecp_mp::generator::tff_nose_run::behaviour_specification ,
    ecp_mp::generator::tff_nose_run::behaviour_specification_data_type(false,
    false, true, false, false, false),
    lib::irp6p_m::ROBOT_NAME); //uruchomienie regulatora dla
    trzeciej współrzędnej
16 wait_for_task_termination(false, lib::irp6p_m::ROBOT_NAME,
    lib::irp6p_m::ROBOT_NAME);
17
18 sr_ecp_msg->message("KONIEC programu");
19 }

```

Wydruk 4.1: funkcja `main_task_algorithm` zadania pierwszego

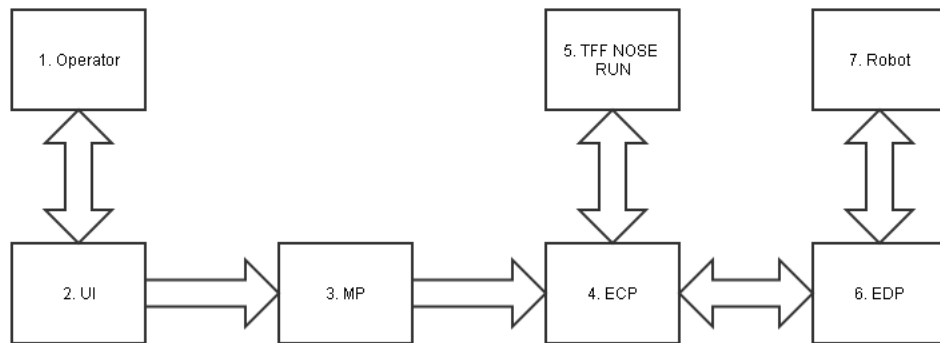
Linie 3, 8, 13 i 18 są informacjami, wypisywanymi na konsolę, aby operator wiedział, jaki jest obecny moment działania programu. Linie 5-6 są rozkazem dla robota, aby używając generatora SMOOTH wykonał trajektorię zapisaną w pliku *trajectory\_postument\_joint.trj*. W liniach 10-11 znajduje się rozkaz wyzerowania odczytów z czujnika sił (generator BIAS\_EDP\_FORCE powoduje uznanie obecnych odczytów za zerowe, od następnych odczytów odejmowane są wartości z tej chwili). W liniach 15-16 znajduje się rozkaz uruchomienia generatora TFF\_NOSE\_RUN dla trzeciej współrzędnej (w pozostałych ma być utrzymywana obecna pozycja).

Schemat blokowy programu używanego w tym celu znajduje się na rysunku 4.1.

Oznaczenia na rysunku to:

1. Operator, który uruchamia zadanie,
2. Proces UI, który pozwala operatorowi na komunikację z procesem MP, prze-





Rysunek 4.1: Schemat programu wykorzystującego regulator

kazuje dane zwrotne i pozwala na włączenie pomiarów,

3. Proces MP, który definiował kolejność wykonywania akcji przez robota,
4. Standardowy proces ECP przekazujący polecenia z MP do generatorów i wywołujący ruch robota,
5. Regulator, dla którego dobierane były parametry,
6. Proces EDP odpowiadający za komunikację z robotem,
7. Robot wykonujący zadanie.

Proces MP został stworzony na cele tej pracy, jako część zależna od zadania. Proces ECP nie został stworzony od nowa, ponieważ standardowy był wystarczający - przekazywał rozkazy ruchu do generatorów, które przekazywały te polecenia sterownikowi, w którym zaimplementowany był regulator sterujący robotem. Dla użytego regulatora dobierane były parametry, ale został on stworzony w ramach struktury MRROC++ przed powstaniem tej pracy.

## 4.2 Zadanie drugie

W drugim zadaniu proces MP realizował przemieszczenie robota do odpowiedniej pozycji, oraz uruchomienia regulatora, który zadawał prędkość w postaci  $\sin(t^2)$ .

### 4.2.1 Zadanie w MRROC++

Na wydruku 4.3. znajduje się funkcja `main_task_algorithm` procesu MP dla tego zadania:

```

1 void askubis_demo::main_task_algorithm(void)
2 {
3   sr_ecp_msg->message("Ruch do pozycji FRONT");
4
5   set_next_ecp_state(ecp_mp::generator::ECP_GEN_SMOOTH_JOINT_FILE_FROM_MP,
6                       1,
7                       "../src/application/askubis_demo_irp6/trajectory_postument_joint.trj",
8                       lib::irp6p_m::ROBOT_NAME);
9
10  wait_for_task_termination(false, lib::irp6p_m::ROBOT_NAME,
11                              lib::irp6p_m::ROBOT_NAME);
12
13  sr_ecp_msg->message("Uruchomienie regulatora");
14
15  set_next_ecp_state(ecp_mp::generator::ECP_GEN_SINUSOIDAL_VELOCITY,
16                    0, "", lib::irp6p_m::ROBOT_NAME); //uruchomienie generatora
17               sinusoidalnej prędkości dla trzeciej współrzędnej.
18  wait_for_task_termination(false, lib::irp6p_m::ROBOT_NAME,
19                              lib::irp6p_m::ROBOT_NAME);
20
21  sr_ecp_msg->message("KONIEC programu");
22 }

```

Wydruk 4.2: funkcja main\_task\_algorithm zadania drugiego

Linie 3, 8 i 13 są informacjami, wypisywanymi na konsolę, aby operator wiedział, w jaki jest obecny moment działania programu. Linie 5-6 są rozkazem dla robota, aby używając generatora SMOOTH wykonał trajektorię zapisaną w pliku *trajectory\_postument\_joint.trj*. W liniach 10-11 znajduje się rozkaz uruchomienia generatora SINUSOIDAL\_VELOCITY, który dla trzeciej współrzędnej generuje trajektorię sinusoidy zwiększającej częstotliwość (w pozostałych ma być utrzymywana obecna pozycja).

W przypadku tego eksperymentu został stworzony nowy proces MP, zaś standardowy proces ECP został zmodyfikowany. Generator prędkości sinusoidalnej też został stworzony i przetestowany na cele tej pracy magisterskiej.

W przypadku procesu ECP do standardowej wersji dodana została w konstruktorze linia odpowiadająca za rejestrację nowego generatora:

```

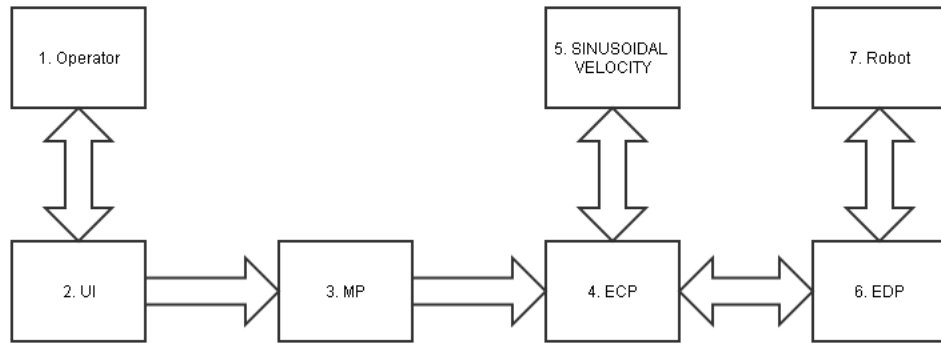
1 register_generator(new generator::sinusoidal_velocity(*this,
2               0));

```

Wydruk 4.3: funkcja main\_task\_algorithm zadania drugiego

Użyty generator ruchu został opisany w rozdziale 4.2.2. Schemat blokowy tego programu znajduje się na rysunku 4.2.

Oznaczenia na rysunku to:



Rysunek 4.2: Schemat programu generującego sinusoidalną prędkość

1. Operator, który uruchamia zadanie,
2. Proces UI, który pozwala operatorowi na komunikację z procesem MP, przekazuje dane zwrotne i pozwala na włączenie pomiarów,
3. Proces MP, który definiował kolejność wykonywania akcji przez robota,
4. Proces ECP rozwinięty o rejestrację w systemie nowego generatora,
5. Generator, który odpowiadał za stworzenie ruchu o prędkości opisanej wzorem  $\sin(t^2)$ ,
6. Proces EDP odpowiadający za komunikację z robotem,
7. Robot wykonujący zadanie.

Hierarchia klas stworzonych modułów na potrzeby tego zadania znajduje się na rysunku 4.3. Klasy najniższe w hierarchii zostały stworzone przeze mnie w celu zrealizowania tego zadania.

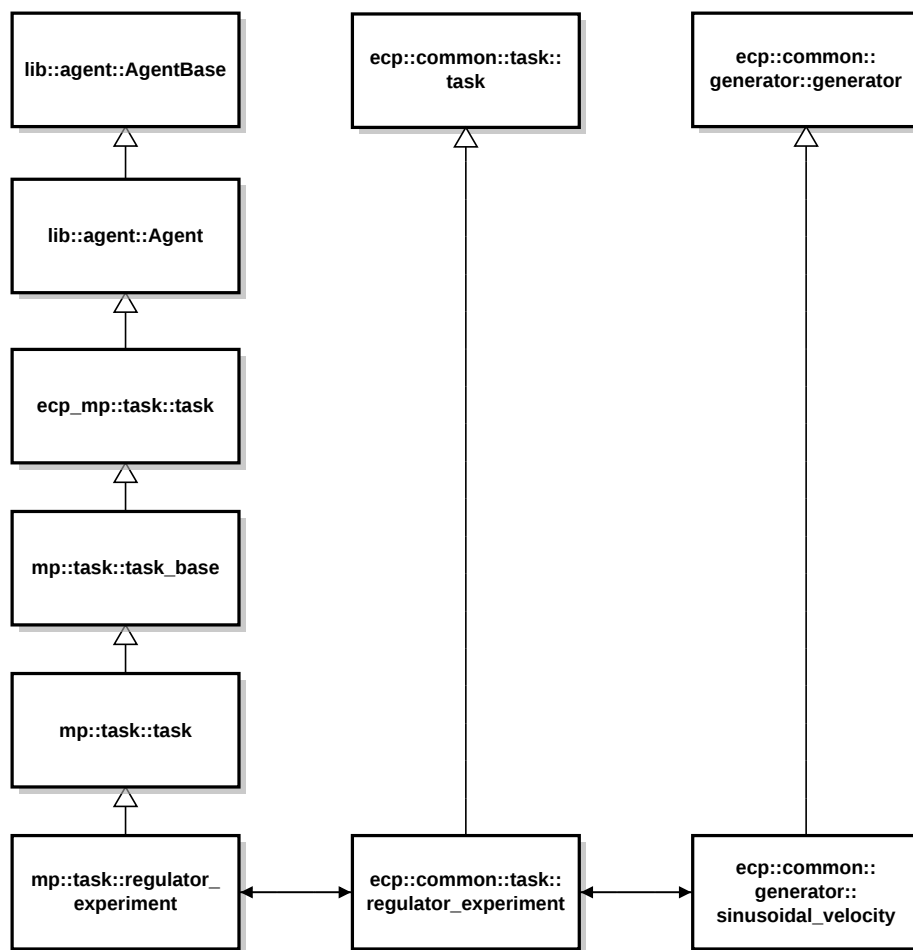
#### 4.2.2 Generator SINUSOIDAL VELOCITY

W celu wykonania eksperymentu 4. stworzyłem dodatkowy generator, mający na celu uzyskanie prędkości w postaci  $\sin(t^2)$ . Zostało to zrealizowane poprzez zmianę kolejnych pozycji zadanych zgodnie ze wzorem:

$$X_d^{t+1} = X_d^t + A * \sin(B * 2 * \pi * t^2) \quad (4.1)$$

gdzie:

- $X_d^{t+1}$  - pozycja zadana w chwili następnej,



Rysunek 4.3: Diagram klas stworzonych do zadania drugiego

- $X_d^t$  - pozycja zadana w chwili obecnej,
- A - stała amplitudy,
- B - stała czasu.

Kod generujący ten ruch został podzielony według standardów obowiązujących w systemie MRROC++ na dwie główne funkcje:

- `first_step`, wywoływana raz, na początku działania generatora, mającą na celu określenie początkowej pozycji robota jako pozycji wyjściowej,
- `next_step`, wywoływana co makrokrok (5 mikrokroków, 10 ms), mającą na celu obliczenie nowej pozycji i wysłanie jej do procesu EDP.

Kod funkcji `first_step` znajduje się na wydruku 4.4. W tej funkcji pobierana jest aktualna wartość pozycji robota, która jest zapisywana i używana jako odniesienie do przyszłych ruchów.

```

1 | bool askubis_sinusoidal_velocity::first_step()
2 | {
3 |   lib::Homog_matrix actual_position_matrix;
4 |   actual_position_matrix =
      the_robot->reply_package.arm.pf_def.arm_frame; //pobranie
      aktualnej pozycji robota
5 |   lib::Xyz_Euler_Zyz_vector euler_vector;
6 |   actual_position_matrix.get_xyz_euler_zyz(euler_vector);
7 |   first_pos = euler_vector; //zapisanie jej jako pozycji startowej
8 |   current_pos = euler_vector[0]; //oraz jako pozycji obecnej
      zadanej
9 |   return true; //sygnał, że funkcja zakończyła się pomyślnie
10 | }
```

Wydruk 4.4: funkcja `first_step`

Linia 4 oznacza pobranie aktualnej pozycji robota. W linii 6 jest ona przetwarzana na współrzędne XYZ EULER ZYZ. W liniach 7-8 ta pozycja jest zapisywana jako startowa i obecna zadana.

Kod funkcji `next_step` znajduje się na wydruku 4.5. W tej funkcji liczona jest zmiana pozycji, która jest dodawana do obecnej zadanej pozycji i wysyłana do kontrolera robota jako nowa zadana prędkość.

```

1 | bool askubis_sinusoidal_velocity::next_step()
2 | {
3 |   step++; //inkrementacja kroku algorytmu
4 |   double time = step*0.01;
5 |   double A= 0.001; //stała amplitudy
6 |   double B = 0.5; //stała czasu
7 |   double gain = A*std::sin(2*3.14*B*time*time); //obliczenie
      zmiany pozycji (zadanej prędkości)
8 |   current_pos+=gain; //zmiana pozycji
9 |   mrrocpp::lib::Xyz_Euler_Zyz_vector new_pos =
      first_pos; //skopiowanie pierwszej pozycji do nowej pozycji
10 |   new_pos[0]=current_pos; //wpisanie nowej wartości pozycji do
      wektora, który zostanie wysłany do robota
11 |   the_robot->communicate_with_edp = true; //włączenie komunikacji
      ze sterownikiem
12 |   the_robot->ecp_command.instruction_type =
      lib::SET; //zaznaczenie, że dane będą wysyłane do sterownika
13 |   the_robot->ecp_command.arm.pf_def.arm_frame.set_from_xyz_euler_zyz(new_pos);
      //zapisanie nowoobliczonej pozycji jako zadanej
```

```
14 | }
```

Wydruk 4.5: funkcja next\_step

W liniach 3-6 zdefiniowane są zmienne użyte do obliczenia zmiany pozycji. Jest ona liczona w liniach 7-8. W liniach 11-13 nowa pozycja jest wysyłana do sterownika robota.

### 4.3 Programy symulacyjno-obliczeniowe w środowisku MATLAB

Program symulacyjny napisany w środowisku MATLAB składał się z dwóch głównych części:

- Programu symulacyjnego w pakiecie SIMULINK przedstawionego na rysunku 2.4,
- Funkcji, która odpowiadała za ustalenie wartości parametrów.

Funkcja ustalająca wartości parametrów przedstawiona jest na wydruku 4.6.

```
1 | Bp=800;
2 | Mp=0.07;%Parametry sterownika
3 | M=1.3;%Parametry układu
4 | B=0;
5 | K=0;
6 | Ta = 0.0103;%Parametry robota
7 | Tb = 0.0025;
8 | Td = 0.0245;
9 | Tab=Ta*Tb;%przeliczenie parametrów dla symulacji
10| TAB=Ta+Tb;
```

Wydruk 4.6: funkcja ustawiająca parametry dla symulacji

Parametry w liniach 1-2 opisują regulator. Linie 3-5 opisują parametry narzędzia. W liniach 6-8 znajdują się nieznane parametry systemu robotycznego, zaś w liniach 9-10 są one przeliczone na parametry używane przez symulację w programie SIMULINK.

W celu dobierania parametrów wartość bazowa była mnożona przez pewną liczbę całkowitą i przez liczbę losową z zakresu (0-1) i dodanie innej liczby aby przetestować wartości z pewnego zakresu jak na wydruku 4.7.

```
1 | Bp = 500 + (500 * rand());%Bp z zakresu 500-1000
```

Wydruk 4.7: modyfikacja parametrów do testów

W celu analizy danych symulacyjnych przez inne programy napisany został program, który uruchamiał symulację, zbierał z niej dane i zapisywał do pliku. Znajduje się on na wydruku 4.8.

```

1  simOut = sim('force.mdl', 'ReturnWorkspaceOutputs','on');
    %uruchomienie symulacji za zwracaniem danych wyjściowych
2  yout = simOut.get('yout');%zwracane dane
3  tout = simOut.get('tout');%znaczniki czasowe do zwracanych
    danych
4  Fm_ = yout(:,1);%wstępne wektory danych, nieskrócone
5  Am_ = yout(:,2);
6  Vm_ = yout(:,3);
7  Vd_ = yout(:,4);
8  t_vec=[];
9  Fm=[];
10 Am=[];
11 Vm=[];
12 Vd=[];
13
14 %skracanie wektorów do długości 5000 próbek (10s przy T =
    0.002s)
15 t=0.0;
16 for i= 2:size(Fm_)
17     if tout(i)>=t
18         t=t+0.002;
19         t_vec=[t_vec;tout(i)];
20         Fm=[Fm;Fm_(i)];
21         Am=[Am;Am_(i)];
22         Vm=[Vm;Vm_(i)];
23         Vd=[Vd;Vd_(i)];
24     end
25 end
26
27 fid = fopen('dla regulatora.txt', 'a');%otwarcie pliku
28 fprintf(fid, 'Mp %20.9f Bp %20.9f\n', Mp, Bp);%zapisanie
    istotnych(testowanych) parametrów symulacji
29 fprintf(fid, 'Fm, Am, Vm, Vd\n');
30 for i= 1:size(Fm)%zapisanie wektorów wynikowych
31     fprintf (fid, '%20.9f %20.9f %20.9f %20.9f\n', Fm(i),
        Am(i), Vm(i), Vd(i));
32 end

```

Wydruk 4.8: funkcja uruchamiająca symulację i zapisująca jej wyniki do pliku

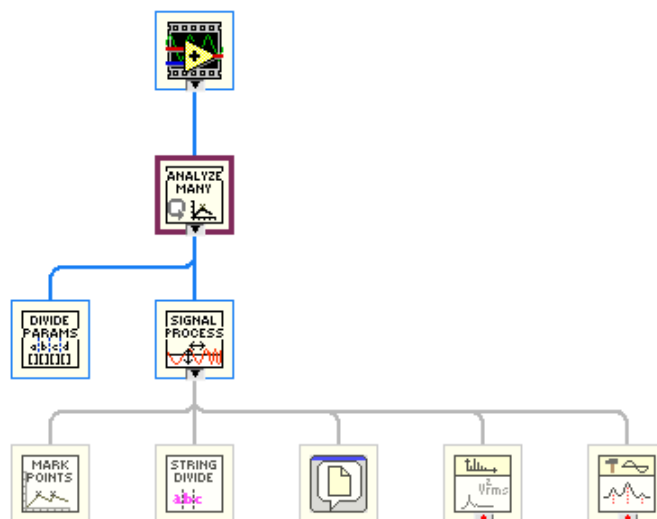
Linia 1 jest uruchomieniem symulacji. Linie 2-3 służą odzyskaniu ze zwracanych danych wektorów danych wyjściowych i znaczników czasowych. W liniach 4-12 two-

rzony są wektory, w których będą się znajdować wartości parametrów. W liniach 14-25 wektory wynikowe są skracane, tak aby dane wynikowe były podobne do danych zwracanych z systemu robotycznego (pomiar co 0.002 s). W liniach 27-31 dane są zapisywane do pliku razem z parametrami, dla których była uruchamiana symulacja.

## 4.4 Program obliczeniowy w LabVIEW

Program stworzony w środowisku LabVIEW miał na celu analizę danych zapisanych w środowisku MATLAB i pobranych z robota - obliczenie częstotliwości rezonansowej i wzmocnień według wzorów z rozdziału 5.2.1. W tym celu została stworzona seria programów, które służyły badaniu danych.

Schemat zależności funkcji znajduje się na rysunku 4.4.

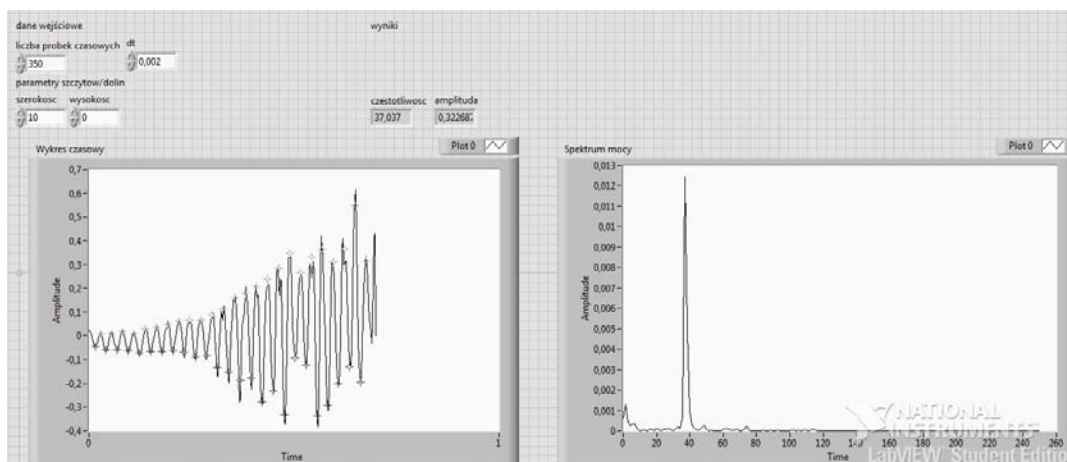


Rysunek 4.4: Hierarchia użytych programów w LabVIEW

Funkcja Analize Many ma na celu przetworzenie bloku danych stworzonych w programie MATLAB. Czytany plik jest dzielony na pojedyncze zestawy danych (z jednej symulacji). Te dane są dzielone na nagłówek zawierający wartości badanych parametrów, który jest przepisywany do pliku wynikowego, oraz wartości zapisanych sygnałów. Wartości sygnałów są następnie separowane od siebie i przetwarzane na tablice przy użyciu funkcji Divide Params. Dla każdego z tych sygnałów wywoływana jest funkcja Signal Process. Z otrzymanych amplitud wyliczane jest wzmocnienie, następnie zapisywane jest ono do pliku wynikowego wraz z częstotliwościami sygnałów.

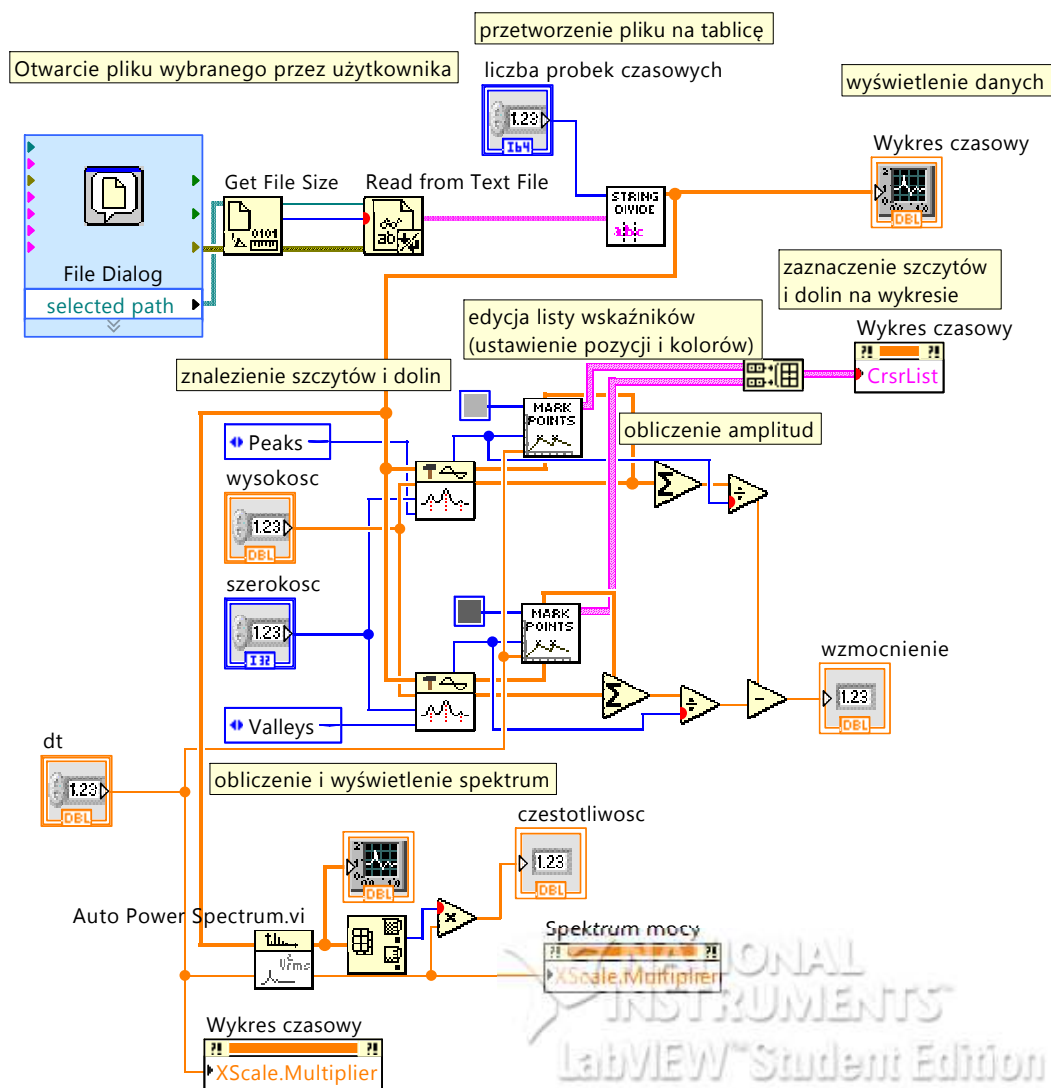


Funkcja Signal Process może przyjmować jako wejście tablicę z wartościami sygnału lub pobierać je ze wskazanego przez użytkownika pliku. Diagram blokowy i panel czołowy tej funkcji przedstawione są na rysunkach 4.5. i 4.6. Na panelu czołowym znajdują się parametry wejściowe - liczba próbek sygnału, kwant czasu pomiędzy pomiarami, minimalna szerokość i wysokość szczytów. Pokazane są też wyniki działania programu - częstotliwość sygnału, jego amplituda, wykres czasowy z zaznaczonymi szczytami i dolinami, oraz charakterystyka widmowa sygnału. Ta funkcja przy użyciu podfunkcji String Divide przetwarza dane odczytane z pliku na tablicę i przedstawia je na diagramie. Przy użyciu transformaty Fouriera (funkcja biblioteczna Auto Power Spectrum) liczone jest spektrum sygnału i wyznaczana jest częstotliwość dominująca (uznawana za częstotliwość drgań). Przy użyciu funkcji Peak Detector znajdują się szczyty i doliny funkcji, następnie liczona jest ich średnia, a różnica pomiędzy średnią wartością dla wierzchołków i dolin uznana została za amplitudę sygnału. W celu zaznaczenia znalezionych dolin i szczytów na wykresie czasowym została stworzona funkcja Mark Points, która dla zadanej listy pozycji, odstępu czasowego pomiędzy pomiarami i koloru zwraca listę kursorów (znaczników). Te kursory umieszczane są na wykresie.



Rysunek 4.5: Panel czołowy (UI) funkcji Signal Process

Pozostałe funkcje znajdujące się na rysunku 4.4. są funkcjami bibliotecznymi służącymi do wyboru pliku przez użytkownika, obliczenia charakterystyki częstotliwościowej sygnału oraz znajdowania szczytów i dolin sygnału.



Rysunek 4.6: Diagram blokowy funkcji Signal Process

## Rozdział 5

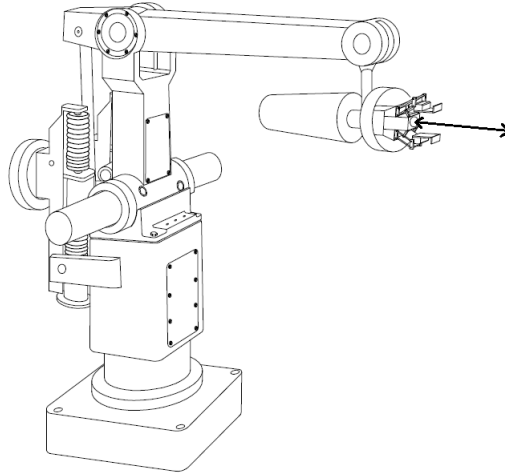
# Identyfikacja parametrów układu

Eksperymenty przedstawione w tym rozdziale miały na celu weryfikację możliwości i zysków płynących z użycia jednostki inercyjnej w sterowaniu siłowym. Z literatury [2, 3] wiadomo, że dla częstotliwości do ok. 10 Hz wykorzystanie jednostki inercyjnej powoduje poprawę pomiarów sił.

W rozdziale 5.1. przedstawiono budowę stanowiska badawczego, na którym zostały przeprowadzone eksperymenty. Eksperyment 1. przedstawiony w rozdziale 5.2. polegał na wzbudzeniu robota do rezonansu i zbadaniu parametrów tego zachowania. Następnie poprzez symulację w pakiecie SIMULINK dobrane zostały parametry, dla których zachowanie symulowanego i rzeczywistego układu robotycznego było możliwie bliskie. Eksperyment 2. przedstawiony w rozdziale 5.3. miał na celu zbadanie parametrów w ten sam sposób, co eksperyment 1., jednak z większym tłumieniem regulatora. Eksperyment 3. przedstawiony w rozdziale 5.4. miał na celu zbadanie parametrów w ten sam sposób, co eksperyment 1., jednak z dołączoną dodatkową masą do końcówki robota. Eksperyment 4. przedstawiony w rozdziale 5.5. miał na celu zbadanie parametrów w ten sam sposób, co eksperyment 1., jednak z zadawanym dodatkowym tłumieniem. Eksperyment 5. przedstawiony w rozdziale 5.6. miał na celu weryfikację hipotezy o wyższej częstotliwości rezonansowej układu z IMU i bez IMU. Polegał na przeprowadzeniu eksperymentu jak w eksperymentcie 1. z tak zmienionym tłumieniem regulatora, aby układ z IMU nie wzbudzał się, zaś układ bez IMU wpadał w samoczynne drgania. Eksperyment 6. przedstawiony w rozdziale 5.7. miał na celu przeprowadzenie eksperymentu jak w eksperymentcie 5., ale poprzez zmianę inercji. W rozdziale 5.8. znajduje się podsumowanie eksperymentów.

## 5.1 Budowa stanowiska badawczego

Stanowisko, na którym przeprowadzane były badania znajduje się w laboratorium robotyki Instytutu Automatyki i Informatyki Stosowanej Politechniki Warszawskiej. Do przeprowadzania eksperymentów używany był robot IRp-6, opisany w rozdziale 3.1. Eksperymenty zostały przeprowadzone dla pojedynczej osi robota w kierunku równoległym do podłoża jak na rysunku 5.1.



Rysunek 5.1: Stanowisko eksperymentalne

Na manipulator nie działały żadne zewnętrzne siły, poza siłą mającą na celu wzbudzenie drgań robota (nieobecną gdy robot już sam utrzymuje drgania), oraz tłumieniem zadawanym w eksperymencie 4.

W ramach eksperymentu rejestrowane były następujące sygnały:

- Zadana prędkość końcówki robota  $\dot{X}_d$ ,
- Zmierzona siła  $F_m$ ,
- Zmierzona prędkość końcówki robota  $\dot{X}_m$ ,
- Zmierzone przyspieszenie końcówki robota  $\ddot{X}_m$ .

## 5.2 Eksperyment 1.

Eksperyment miał na celu zbadanie częstotliwości rezonansowej układu, oraz wzmocnień członów  $G_2$  i  $G_{3I4}$  w celu wyznaczenia wartości parametrów układu robotycznego w oparciu o metodykę zaprezentowaną w [2].

### 5.2.1 Sposób przeprowadzenia eksperymentu

Przeprowadzenie eksperymentu polegało na zadaniu układowi robotycznemu siły, którą on starał się niwelować (takie polecenie zostało wydane sterownikowi). Układ po zakończeniu przykładania siły zaczynał sam wzbudzać swoje drgania rezonansowe.

Celem tego eksperymentu było wyznaczenie następujących parametrów:

- $f_{re}$  można otrzymać ze wzoru:

$$f_{re} = n_0/T_0 \quad (5.1)$$

gdzie  $n_0$  jest liczbą okresów, a  $T_0$  jest okresem czasu, w którym te drgania występują. Obie te wartości można uzyskać z wykresu siły od czasu. W celu poprawienia jakości wyznaczania tego parametru jego wartość otrzymywana jest przy użyciu transformaty Fouriera z widma częstotliwości,

- Funkcja wzmocnienia  $L_e(G_2)$  uzyskiwana jest ze wzoru:

$$L_e(G_2) = 20 \log A(\dot{X}_m)/A(\dot{X}_d) \quad (5.2)$$

gdzie  $A(\dot{X}_m)$  jest średnią amplitudą sygnału  $\dot{X}_m$  podczas oscylacji,  $A(\dot{X}_d)$  analogicznie,

- Funkcja przesunięcia  $P_e(G_2)$  uzyskiwana jest ze wzoru:

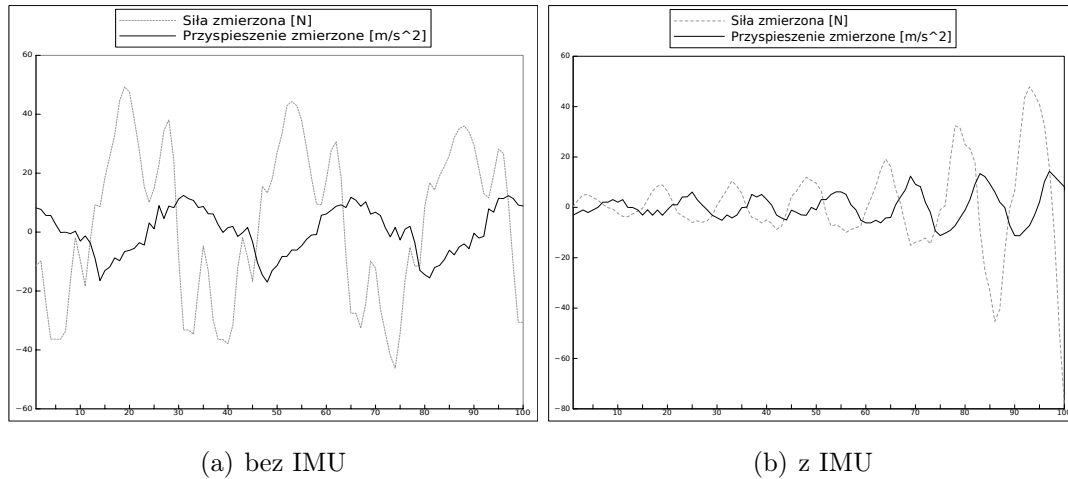
$$P_e(G_2) = 360(T_s/T_p) \quad (5.3)$$

gdzie  $T_s$  jest średnim przesunięciem fazowym pomiędzy sygnałami  $\dot{X}_d$  i  $\dot{X}_m$ , a  $T_p$  średnim okresem oscylacji,

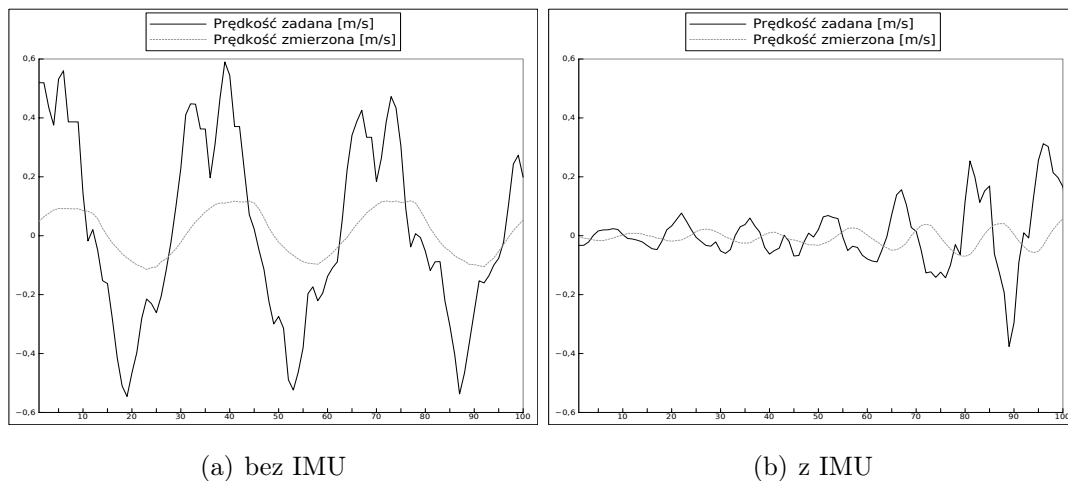
- Funkcje  $L_e(G_{3I4})$  i  $P_e(G_{3I4})$  uzyskiwane są analogicznie do  $L_e(G_2)$  i  $P_e(G_2)$  na podstawie sygnałów  $\ddot{X}_m$  i  $F_m$ .

### 5.2.2 Wyniki eksperymentu

Na rysunkach 5.2. i 5.3. znajdują się wykresy wyników eksperymentów bez użycia jednostki inercyjnej, oraz z użyciem jednostki inercyjnej. Wykresy te przedstawiają tylko drgania sygnału, nie został na nich przedstawiony proces wzbudzania. Charakterystyczną cechą wszystkich wykresów z eksperymentów przeprowadzanych na rzeczywistym obiekcie jest zauważalna niedokładność - sygnał nie ma postaci idealnej sinusoidy, lecz jest poszarpany i nierówny. W tym eksperymencie można zauważyć, że amplituda sygnału z użyciem IMU rośnie - jest to spowodowane bardzo dużym



Rysunek 5.2: Wykresy siły zmierzonej ( $F_m$ ) i przyspieszenia zmierzonego ( $\ddot{X}_m$ ) dla eksperymentu



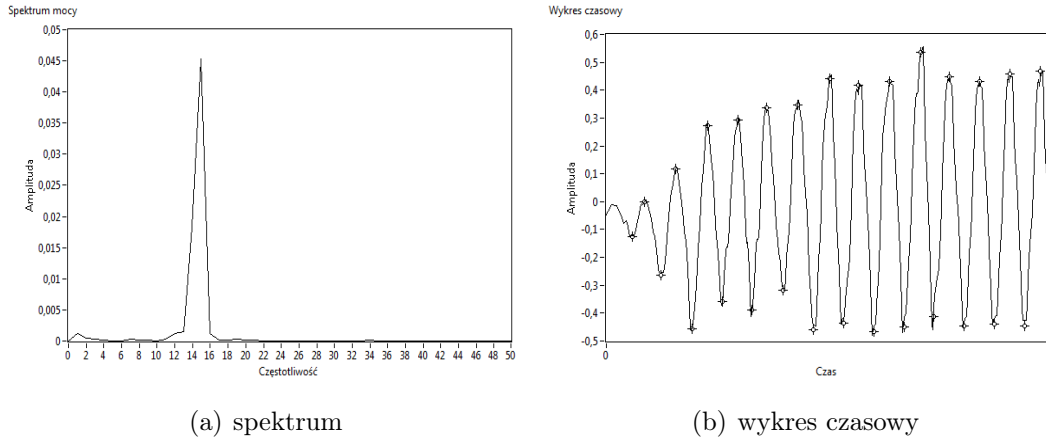
Rysunek 5.3: Wykresy prędkości zmierzonej ( $\dot{X}_m$ ) i zadanej ( $\dot{X}_d$ ) dla eksperymentu

wzmocnieniem - drgania wzbudzały się samoczynnie i bardzo szybko przekraczały dopuszczalne zakresy wartości sił.

Dla zebranych w eksperymencie danych zmierzone zostały częstotliwości sygnałów o najwyższych amplitudach oraz amplituda średnia sygnału (peak to peak). Dane te były analizowane przy pomocy programu LabVIEW, który udostępnia funkcje zarówno analizy częstotliwościowej jak i znajdowania szczytów i dolin sygnału. Dokładniejszy opis programu LabVIEW znajduje się w rozdziale 3.4, a opis zaimplementowanych w nim funkcji w 4.4.

Na rysunku 5.4. znajdują się przykładowe wykresy spektrum mocy(5.4(b)) i w funkcji czasu z zaznaczonymi szczytami i dolinami(5.4(a)) dla pomiaru prędkości zmierzonej ( $\dot{X}_m$ ) bez użycia IMU. Widoczne jest, że wierzchołki nie są znajdowane dokładnie, jednak ponieważ te niedokładności występują na każdym z wykresów,

a interesującym nas parametrem jest stosunek amplitud, uważam że te błędy mają niewielki wpływ na wynik końcowy. Rysunki te mają na celu ukazanie sposobu liczenia parametrów układu i nie będą one powtarzane przy następnych eksperymentach.



Rysunek 5.4: Spektrum mocy i wykres czasowy z zaznaczonymi szczytami i dolinami dla pomiaru prędkości zmierzonej bez wykorzystania IMU

Tabele 5.1. i 5.2. zawierają wyniki analizy sygnałów zebranych w eksperymencie. Można zauważyć, że przy włączonym IMU znacząco zmieniła się częstotliwość drgań rezonansowych. Oznacza to, że dla systemu z włączonym IMU można zaprojektować lepiej działający sterownik, który będzie zawierał człon dolnoprzepustowy, a jednocześnie będzie odcinał znacznie mniejszy zakres częstotliwości.

$f_{re}$	$f_{rs}$	$B_p$	$M_p$	$T_a$	$T_b$	$T_d$	$M$	$B$	$K$
14.98	14.34	70	0	0.00203	0.0296	0.0245	1.3	0	0
				$G_1$		$G_2$		$G_{3I4}$	
				exp, sim		exp	sim	exp	sim
$\mathcal{L}()$				-36.9		-10.92	-10.64	7.41	3.66
$\mathcal{P}()$				0		30	50	130	140

Tablica 5.1: Parametry dla eksperymentu 1. bez IMU

Po zakończeniu analizy eksperymentów należało dobrać nieznane parametry układu:

- Wartości  $B_p$ ,  $M_p$ ,  $M$ ,  $B$  oraz  $K$  są znane,
- Wartość  $T_d$  należy dobrać, aby uzyskać dopasowanie funkcji  $L_e(G_{3I4}) = L_s(G_{3I4})$  i  $f_{re} = f_{rs}$ ,

$f_{re}$	$f_{rs}$	$B_p$	$M_p$	$T_a$	$T_b$	$T_d$	$M$	$B$	$K$
34.65	34.03	70	0	0.00105	0.0136	0.0098	1.3	0	0
				$G_1$		$G_2$		$G_{3I4}$	
				exp, sim		exp	sim	exp	sim
$\mathcal{L}()$				-36.9		-9.35	-10.18	8.40	0.58
$\mathcal{P}()$				0		80	90	100	120

Tablica 5.2: Parametry dla eksperymentu 1. z IMU

- Wartości  $T_a$  i  $T_b$  należy dobrać, aby uzyskać dopasowanie funkcji  $L_e(G_2) = L_s(G_2)$ ,  $P_e(G_2) = P_s(G_2)$  i  $f_{re} = f_{rs}$ .

Dobór powyższych parametrów został zrealizowany przez przetestowanie w programie SIMULINK wielu zestawów parametrów  $T_a$ ,  $T_b$ ,  $T_d$  i wyliczeniu wartości  $f_{rs}$ ,  $L_s(G_2)$  i  $P_s(G_2)$ . Sygnałem zadawanym na wejściu symulacji był sygnał sinusoidalny o zmiennej częstotliwości (chirp signal). Dla tego sygnału znajdowana była częstotliwość o najwyższym wzmocnieniu i dla niej wyliczane były wzmocnienia na podstawie minimalnych i maksymalnych wartości sygnałów. Następnie wybrane zostały wartości parametrów, dla których wynik był bliski wynikom eksperymentów. Dodatkowym warunkiem był wygląd wykresu, ponieważ przy wykresie bardzo oddalonym od docelowego (np. bez oscylacji) znajdowane były pewne częstotliwości dominujące i wzmocnienia sygnałów, które przypadkowo mogły być zgodne z szukanymi. Model robota IRp-6 w środowisku SIMULINK, na którym przeprowadzane były symulacje znajduje się na rysunku 2.4. Wartości otrzymane w ten sposób zostały przedstawione w poprzednich tabelach.

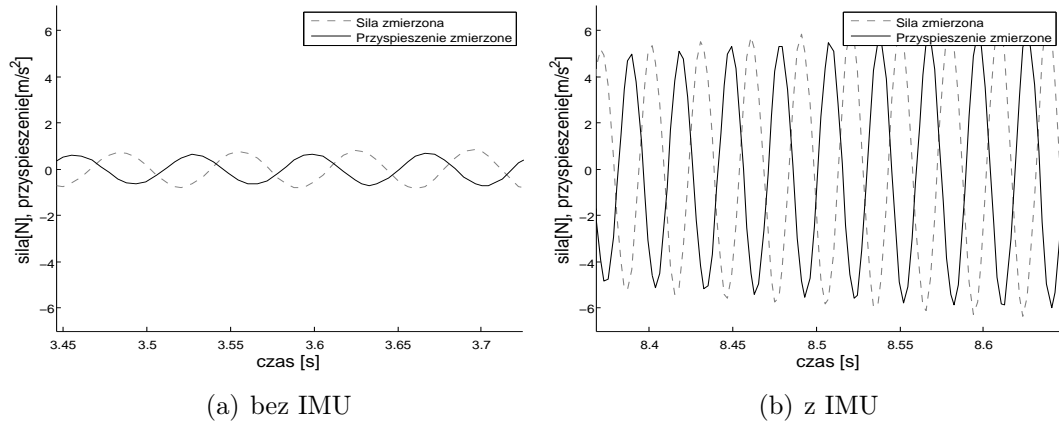
Na rysunku 5.5. znajduje się porównanie sygnałów uzyskanych przy pomocy symulacji - siły zmierzonej i przyspieszenia zmierzonego dla obu uzyskanych zestawów parametrów.

Na rysunku 5.6. znajduje się porównanie wykresów prędkości zadanej i zmierzonej dla symulacji z IMU i bez IMU.

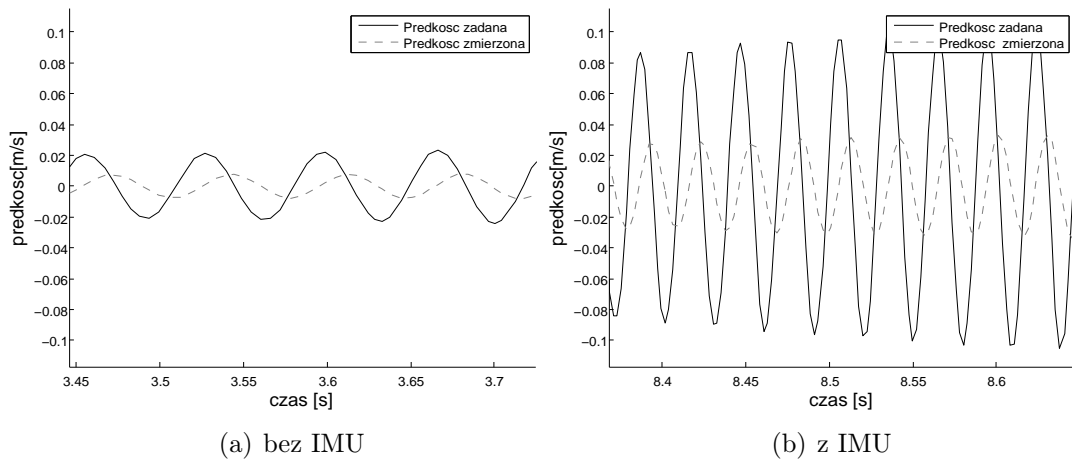
Rysunek 5.7. przedstawia porównanie wykresów Bodego dla układu symulowanego z IMU i bez. Wykresy te wydają się potwierdzać przypuszczenia, że rezonans dla układu z IMU występuje dla wyższej częstotliwości. Pierwszy szczyt wykresu częstotliwości (najbardziej znaczący ze względu na najwyższą amplitudę) występuje dla wyższych częstotliwości niż dla układu bez IMU.

Zauważalna jest też większa amplituda sygnałów w symulacji przy użyciu IMU. W przypadku eksperymentów nie jest to zauważalne z powodu szybkiego przekroczenia ograniczeń przez układ z IMU.





Rysunek 5.5: Wykres siły zmierzonej ( $F_m$ ) i przyspieszenia zmierzonego ( $\ddot{X}_m$ ) z symulacji



Rysunek 5.6: Wykres prędkości zmierzonej ( $\dot{X}_m$ ) i zadanej ( $\dot{X}_d$ ) z symulacji

## 5.3 Eksperyment 2.

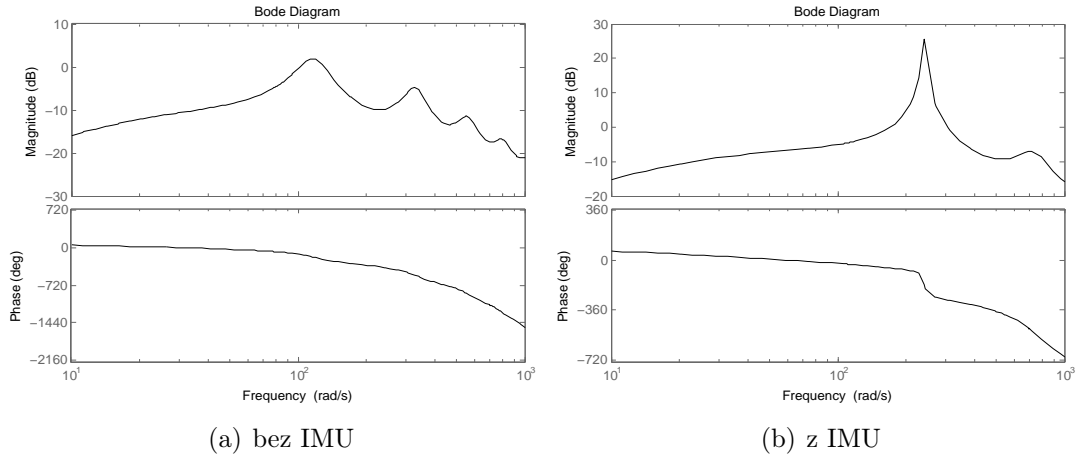
### 5.3.1 Sposób przeprowadzenia eksperymentu

Eksperyment polegał na wykonaniu pomiarów jak w eksperymencie 1. dla układu z wyższym tłumieniem w regulatorze.

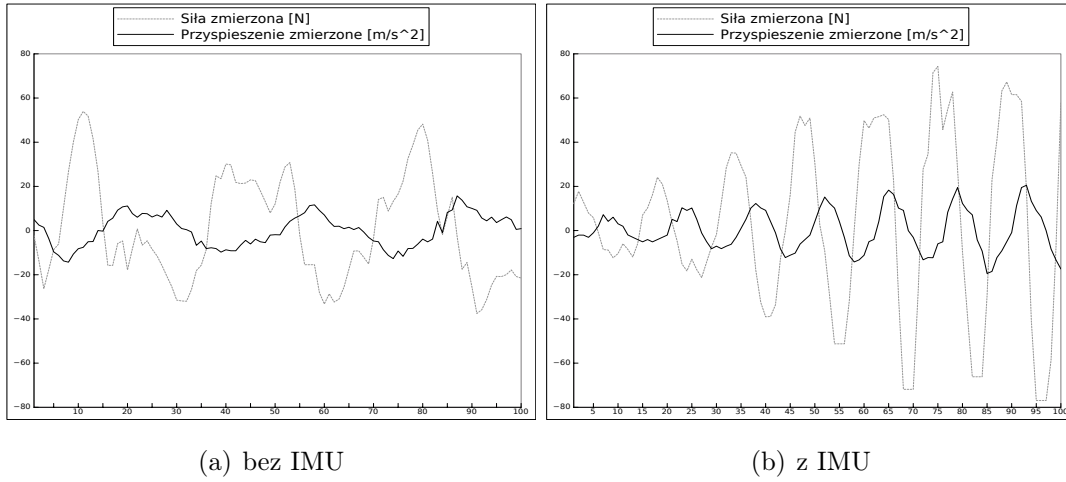
### 5.3.2 Wyniki eksperymentu

Na rysunkach 5.8. i 5.9. znajdują się wykresy wyników eksperymentów bez użycia jednostki inercyjnej, oraz z użyciem jednostki inercyjnej dla eksperymentów z większym tłumieniem regulatora. Wykresy te przedstawiają tylko drgania wykresu, nie został na nich przedstawiony proces wzbudzenia.

Na podstawie tych sygnałów obliczone zostały wartości  $f_{re}$ ,  $L_e(G_2)$ ,  $P_e(G_2)$ ,



Rysunek 5.7: Wykresy Bodego  $G_Z$  układu z pętlą zamkniętą dla parametrów uzyskanych w eksperymencie 1.



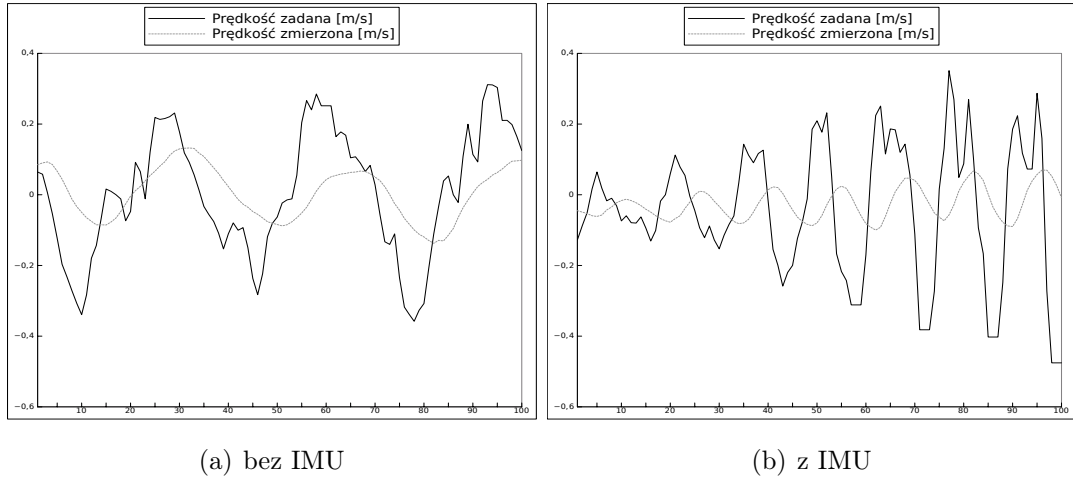
Rysunek 5.8: Wykres siły zmierzonej ( $F_m$ ) i przyspieszenia zmierzonego ( $\ddot{X}_m$ ) dla eksperymentu ze zwiększonym tłumieniem regulatora

$L_e(G_{3I4})$ , oraz  $P_e(G_{3I4})$ . Przedstawione zostały one w tablicach 5.3. oraz 5.4.

W porównaniu do eksperymentu 1.:

- Wzmocnienie członu  $G_2$  zmniejszyło się bez użycia IMU, a zwiększyło się przy użyciu IMU,
- Wzmocnienie członu  $G_{3I4}$  zwiększyło się,
- Częstotliwości rezonansowe układu z IMU i bez IMU pozostały praktycznie bez zmian.

Wykresy siły zadanej i przyspieszenia zadanego uzyskane w wyniku symulacji znajdują się na rysunku 5.10.



Rysunek 5.9: Wykres prędkości zmierzonej ( $\dot{X}_m$ ) i zadanej ( $\dot{X}_d$ ) dla eksperymentu ze zwiększonym tłumieniem regulatora

$f_{re}$	$f_{rs}$	$B_p$	$M_p$	$T_a$	$T_b$	$T_d$	$M$	$B$	$K$
14.66	14.75	100	0	0.00192	0.0256	0.0245	1.3	0	0
				$G_1$		$G_2$		$G_{3I4}$	
				exp, sim		exp	sim	exp	sim
$\mathcal{L}()$				-40		-9.44	-9.49	9.39	3.19
$\mathcal{P}()$				0		40	90	275	250

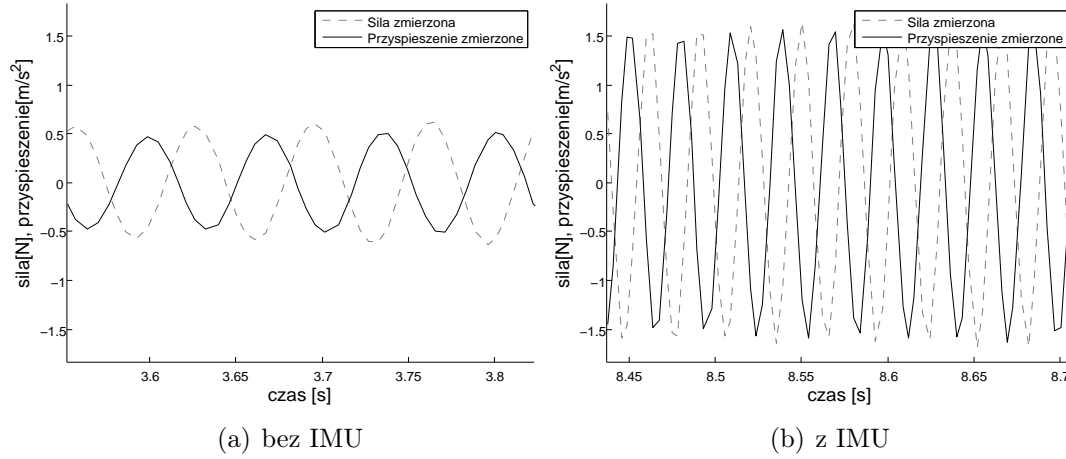
Tablica 5.3: Parametry dla eksperymentu ze zwiększonym tłumieniem regulatora bez IMU

$f_{re}$	$f_{rs}$	$B_p$	$M_p$	$T_a$	$T_b$	$T_d$	$M$	$B$	$K$
34.65	34.29	100	0	0.00213	0.0116	0.0085	1.3	0	0
				$G_1$		$G_2$		$G_{3I4}$	
				exp, sim		exp	sim	exp	sim
$\mathcal{L}()$				-40		-11.50	-9.82	11.95	0.36
$\mathcal{P}()$				0		90	90	200	120

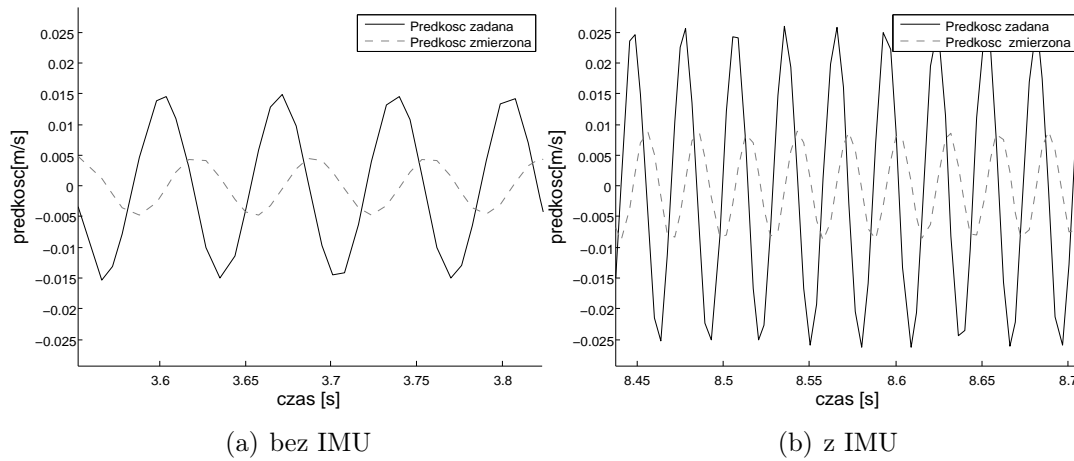
Tablica 5.4: Parametry dla eksperymentu ze zwiększonym tłumieniem regulatora z IMU

Na rysunku 5.11. znajdują się wykresy przedstawiające prędkość zadaną i zmierzona uzyskane w wyniku symulacji dla wartości parametrów uzyskanych w tym eksperymencie.

Na wykresach Bodego dla tych symulacji znajdujących się na rysunku 5.12.



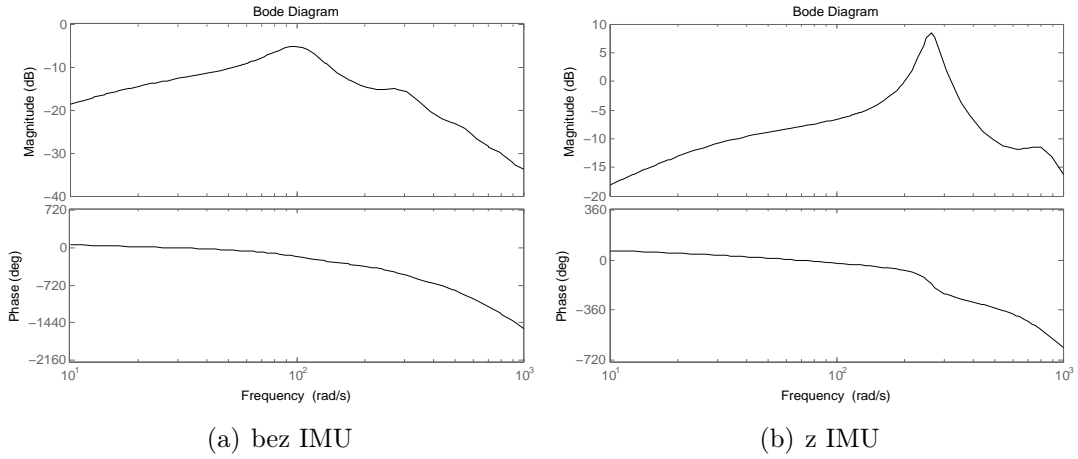
Rysunek 5.10: Wykres siły zmierzonej ( $F_m$ ) i przyspieszenia zmierzonego ( $\ddot{X}_m$ ) z symulacji dla układu ze zwiększonym tłumieniem regulatora



Rysunek 5.11: Wykres prędkości zmierzonej ( $\dot{X}_m$ ) i zadanej ( $\dot{X}_d$ ) z symulacji dla układu ze zwiększonym tłumieniem regulatora

można zauważyć, że są one w miarę zbliżone do tych z eksperymentu 2., jednak szczyty są zdecydowanie mniej ostre.

Wyniki eksperymentu z większym tłumieniem są bardzo zbliżone do eksperymentu 1. Poza drobnymi zmianami wzmocnień i ograniczonym wzmocnieniem dla układu z IMU nie ma innych zauważalnych zmian.



Rysunek 5.12: Wykresy Bodego  $G_Z$  układu z pętlą zamkniętą dla parametrów użytych w eksperymencie ze zwiększonym tłumieniem regulatora

## 5.4 Eksperyment 3.

### 5.4.1 Sposób przeprowadzenia eksperymentu

Eksperyment polegał na wykonaniu pomiarów jak w eksperymencie 1. dla układu z dodatkową masą. W tym celu do chwytaka robota zamontowany został odważnik o masie 1kg. Następnie uzyskane wartości parametrów zostały porównane z tymi z pierwszego eksperymentu.

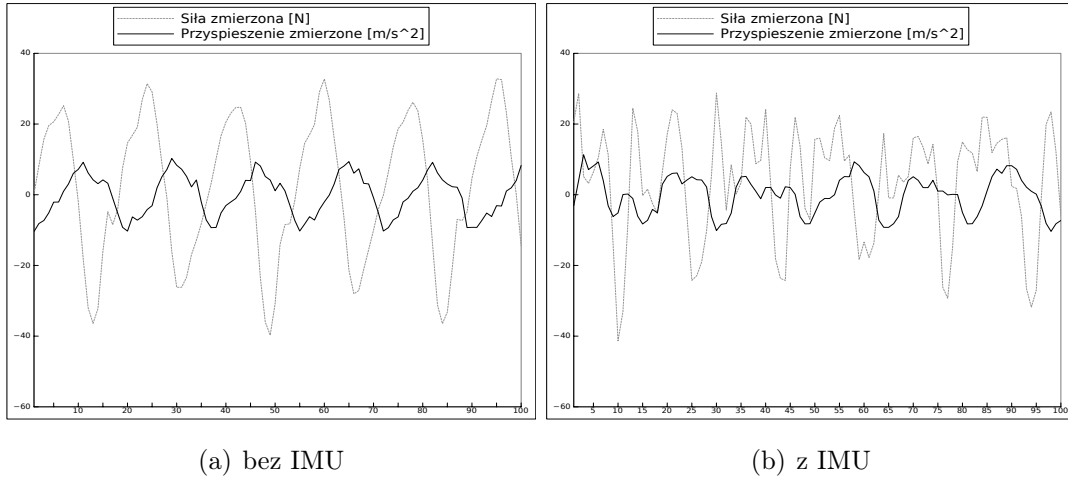
### 5.4.2 Wyniki eksperymentu

Na rysunkach 5.13. i 5.14. znajdują się wykresy wyników eksperymentów bez użycia jednostki inercyjnej, oraz z użyciem jednostki inercyjnej dla eksperymentów z dodatkową masą. Wykresy te przedstawiają tylko drgania wykresu, nie został na nich przedstawiony proces wzbudzenia.

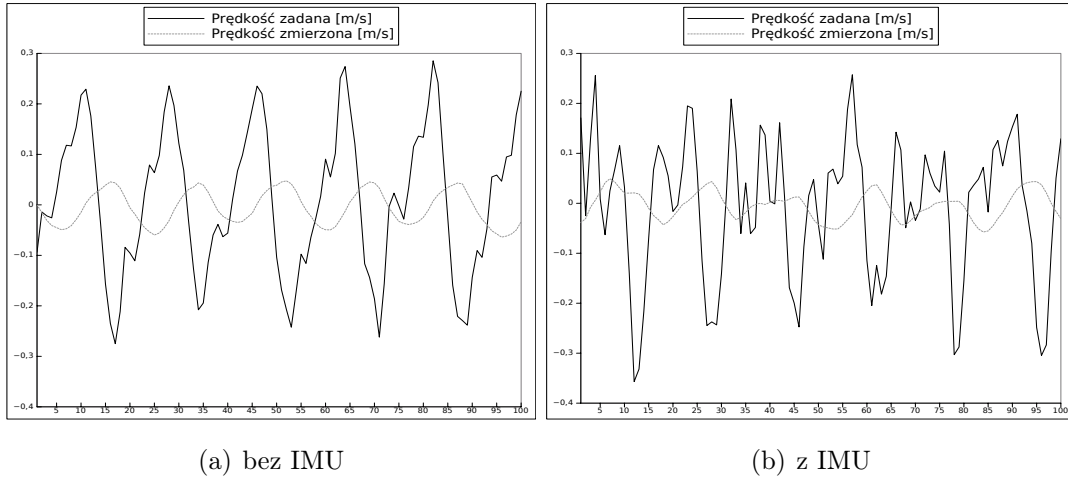
Na podstawie tych wykresów obliczone zostały wartości  $f_{re}$ ,  $L_e(G_2)$ ,  $P_e(G_2)$ ,  $L_e(G_{3I4})$ , oraz  $P_e(G_{3I4})$ . Przedstawione zostały one w tablicach 5.5. oraz 5.6.

W porównaniu do eksperymentu 1.:

- Wzmocnienie członu  $G_2$  z użyciem IMU zmniejszyło się,
- Wzmocnienie członu  $G_{3I4}$  bez użycia IMU zmniejszyło się,
- Częstotliwość rezonansowa układu bez IMU zmieniła się nieznacznie, ale zauważalny jest wzrost mocy częstotliwości harmonicznej (28.47Hz),
- Częstotliwość rezonansowa układu z IMU zmalała.



Rysunek 5.13: Wykres siły zmierzonej ( $F_m$ ) i przyspieszenia zmierzonego ( $\ddot{X}_m$ ) dla eksperymentu z dodatkową masą



Rysunek 5.14: Wykres prędkości zmierzonej ( $\dot{X}_m$ ) i zadanej ( $\dot{X}_d$ ) dla eksperymentu z dodatkową masą

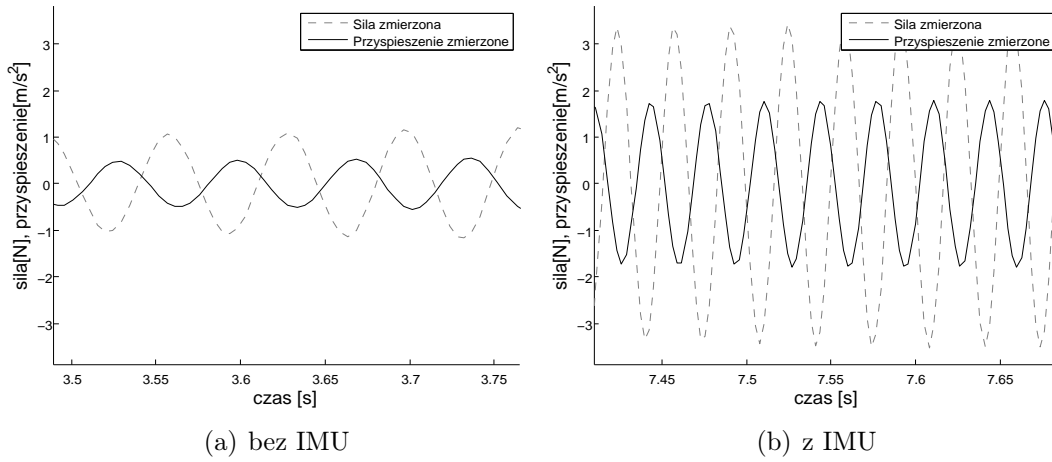
$f_{r_e}$	$f_{r_s}$	$B_p$	$M_p$	$T_a$	$T_b$	$T_d$	$M$	$B$	$K$
14.24	14.51	100	0	0.00155	0.0336	0.0265	2.3	0	0
				$G_1$		$G_2$		$G_{3I4}$	
				exp, sim		exp	sim	exp	sim
$\mathcal{L}()$				-40		-10.96	-11.14	9.53	7.73
$\mathcal{P}()$				0		90	80	240	120

Tablica 5.5: Parametry dla eksperymentu z większą masą bez IMU

Wykresy siły zadanej i przyspieszenia zadanego uzyskane w wyniku symulacji znajdują się na rysunku 5.15.

$f_{r_e}$	$f_{r_s}$	$B_p$	$M_p$	$T_a$	$T_b$	$T_d$	$M$	$B$	$K$
30.63	30.19	100	0	0.00104	0.0206	0.0113	2.3	0	0
				$G_1$		$G_2$		$G_{3I4}$	
				exp, sim		exp	sim	exp	sim
$\mathcal{L}()$				-40		-12.71	-12.39	8.17	5.84
$\mathcal{P}()$				0		150	90	180	110

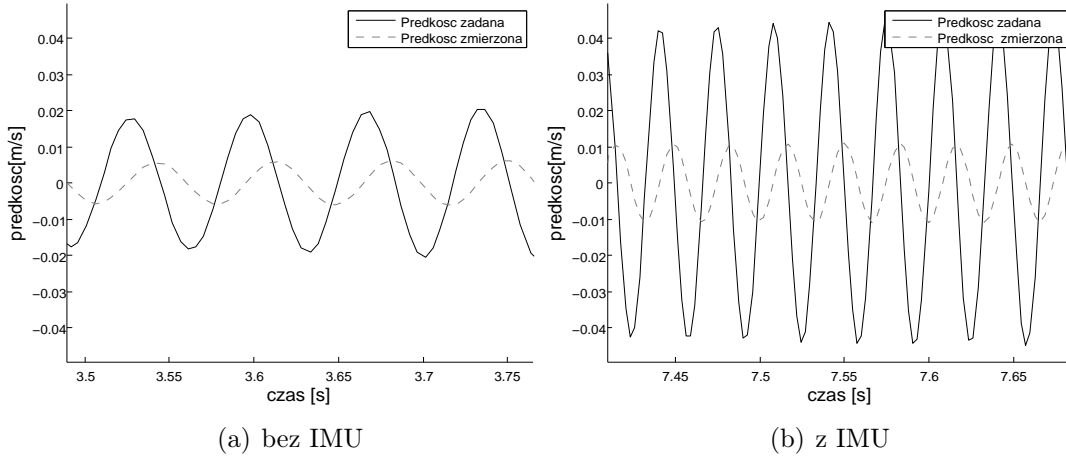
Tablica 5.6: Parametry dla eksperymentu z większą masą z IMU

Rysunek 5.15: Wykres siły zmierzonej ( $F_m$ ) i przyspieszenia zmierzonego ( $\ddot{X}_m$ ) z symulacji dla układu z dodatkową masą

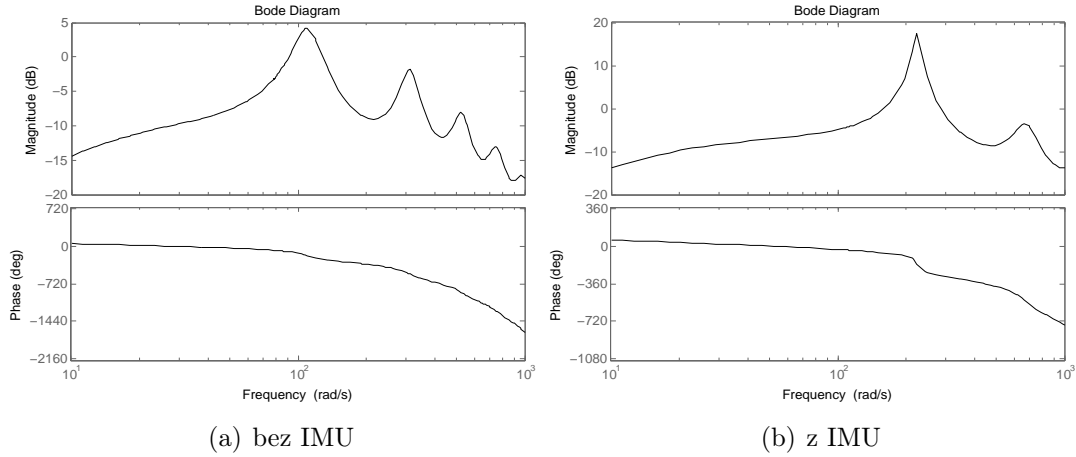
Na rysunku 5.16. znajdują się wykresy przedstawiające prędkość zadaną i zmierzoną uzyskane w wyniku symulacji dla wartości parametrów uzyskanych w tym eksperymencie.

Na wykresach Bodego dla tych symulacji znajdujących się na rysunku 5.17. można zauważyć, że są one do siebie bardzo zbliżone. Zarówno kształt jak i wzmocnienia systemu są do siebie podobne w przypadku układu z IMU i bez IMU. Zgodnie z wynikami eksperymentów zauważalne jest, że częstotliwość drgań układu wzrasta przy włączeniu jednostki inercyjnej. W przypadku eksperymentu z dodatkową masą wzmocnienie transmitancji układu z pętlą zamkniętą jest bardzo duże i posiada bardzo wyraźny (ostry) szczyt rezonansu.

W tym eksperymencie można zauważyć, że wyniki są zbliżone do tych z eksperymentu 1. Zauważalne jest zmniejszenie częstotliwości rezonansowej dla układu z IMU. Przy zwiększeniu masy narzędzia zaobserwować można też zwiększenie wzmocnienia układu.



Rysunek 5.16: Wykres prędkości zmierzonej ( $\dot{X}_m$ ) i zadanej ( $\dot{X}_d$ ) z symulacji dla układu z dodatkową masą



Rysunek 5.17: Wykresy Bodego  $G_Z$  układu z pętlą zamkniętą dla parametrów uzyskanych w eksperymencie z dodatkową masą.

## 5.5 Eksperyment 4.

### 5.5.1 Sposób przeprowadzenia eksperymentu

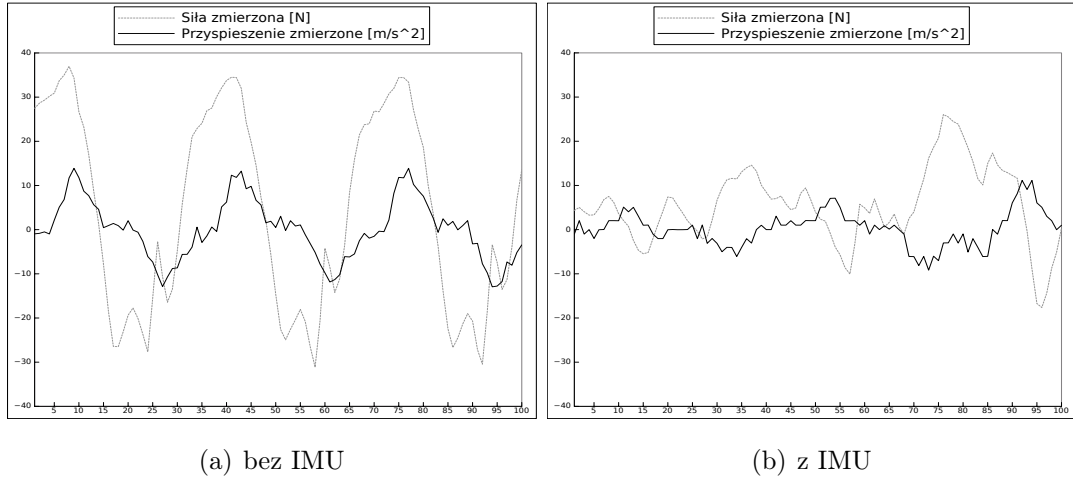
Eksperyment ten polegał na wykonaniu pomiarów jak w eksperymencie 1. dla układu z dodatkowym zadaniem chwytakowi tłumieniem. Następnie uzyskane wartości parametrów miały być porównane z tymi z pierwszego eksperymentu.

### 5.5.2 Wyniki eksperymentu

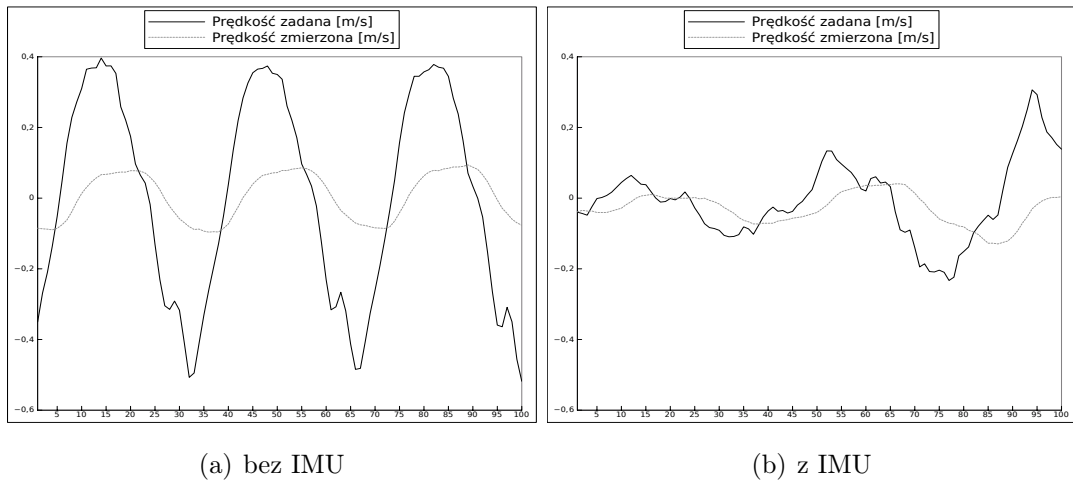
Na rysunkach 5.18. i 5.19. znajdują się wykresy wyników eksperymentów bez użycia jednostki inercyjnej, oraz z użyciem jednostki inercyjnej. Wykresy te przed-



stawiają tylko drgania wykresu, nie został na nich przedstawiony proces wzbudzania.



Rysunek 5.18: Wykres siły zmierzonej ( $F_m$ ) i przyspieszenia zmierzonego ( $\ddot{X}_m$ ) dla eksperymentu z dodatkowym tłumieniem



Rysunek 5.19: Wykres prędkości zmierzonej ( $\dot{X}_m$ ) i zadanej ( $\dot{X}_d$ ) dla eksperymentu z dodatkowym tłumieniem

Na podstawie tych wykresów obliczone zostały wartości  $f_{re}$ ,  $L_e(G_2)$ ,  $P_e(G_2)$ ,  $L_e(G_{3I4})$ , oraz  $P_e(G_{3I4})$ . Przedstawione zostały one w tablicach 5.7. oraz 5.8.

W porównaniu do eksperymentu 1.:

- Wzmocnienia członów  $G_2$  zmalało,
- Wzmocnienie członu  $G_{3I4}$  nieznacznie zmalało,
- Częstotliwość rezonansowa układu bez IMU pozostała bez zmian,
- Częstotliwość rezonansowa układu z IMU zmalała kilkukrotnie.

$f_{r_e}$	$f_{r_s}$	$B_p$	$M_p$	$T_a$	$T_b$	$T_d$	$M$	$B$	$K$
14.97	15.47	100	0	0.00107	0.0451	0.0262	1.3	100	0
				$G_1$		$G_2$		$G_{3I4}$	
				exp, sim		exp	sim	exp	sim
$\mathcal{L}()$				-40		-13.54	-13.26	6.45	3.83
$\mathcal{P}()$				0		45	45	300	120

Tablica 5.7: Parametry dla eksperymentu z dodatkowym tłumieniem bez IMU

$f_{r_e}$	$f_{r_s}$	$B_p$	$M_p$	$T_a$	$T_b$	$T_d$	$M$	$B$	$K$
12.97	13.01	100	0	0.00087	0.0533	0.0332	1.3	100	0
				$G_1$		$G_2$		$G_{3I4}$	
				exp, sim		exp	sim	exp	sim
$\mathcal{L}()$				-40		-13.42	-12.90	7.65	1.92
$\mathcal{P}()$				0		60	90	240	120

Tablica 5.8: Parametry dla eksperymentu z dodatkowym tłumieniem z IMU

Na rysunku 5.20. znajdują się wykresy z symulacji przedstawiające siłę zmierzoną i przyspieszenie zmierzone.

Porównanie wykresów prędkości zadanej i zmierzonej dla układu z IMU i bez znajduje się na rysunku 5.21.

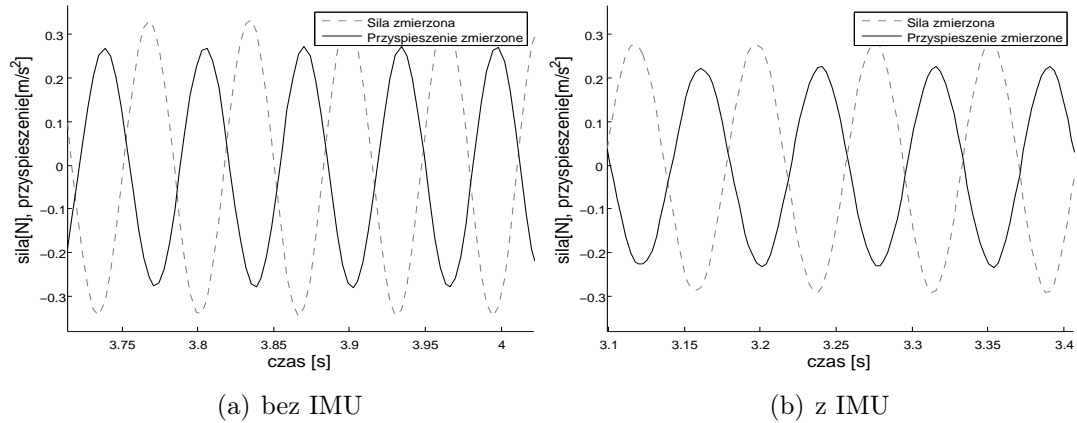
Na rysunku 5.22. znajdują się wykresy Bodego transmitancji  $G_Z$  układu z pętlą zamkniętą dla układu z IMU i bez. Zauważyć można, że w przeciwieństwie do poprzednich przypadków częstotliwość rezonansowa (pierwszy szczyt amplitudy) w obu przypadkach jest w podobnym miejscu. Użycie IMU nie ma dużego wpływu na układ w przypadku istnienia dodatkowego tłumienia.

Przeprowadzone eksperymenty z dodatkowym tłumieniem w układzie ukazują duże zmiany w działaniu systemu. Wzmocnienia sygnałów zmalały, zauważalne jest także większe podobieństwo wyników z użyciem i bez użycia IMU.

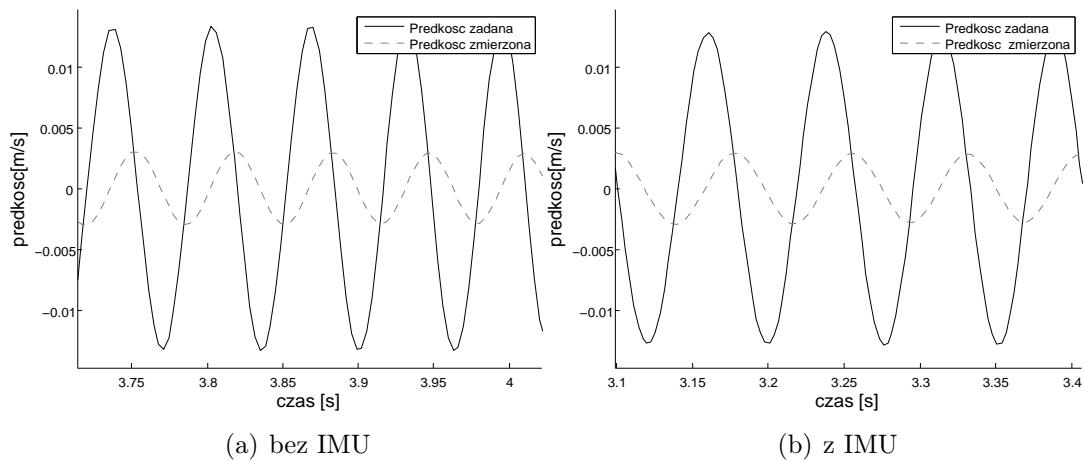
## 5.6 Eksperyment 5.

### 5.6.1 Sposób przeprowadzenia eksperymentu

Eksperyment polegał na wykonaniu pomiarów jak w eksperymencie 1. dla takich parametrów regulatora (poprzez zmianę tłumienia), aby układ bez IMU się wzbudzał



Rysunek 5.20: Wykres siły zmierzonej ( $F_m$ ) i przyspieszenia zmierzonego ( $\ddot{X}_m$ ) z symulacji z dodatkowym tłumieniem



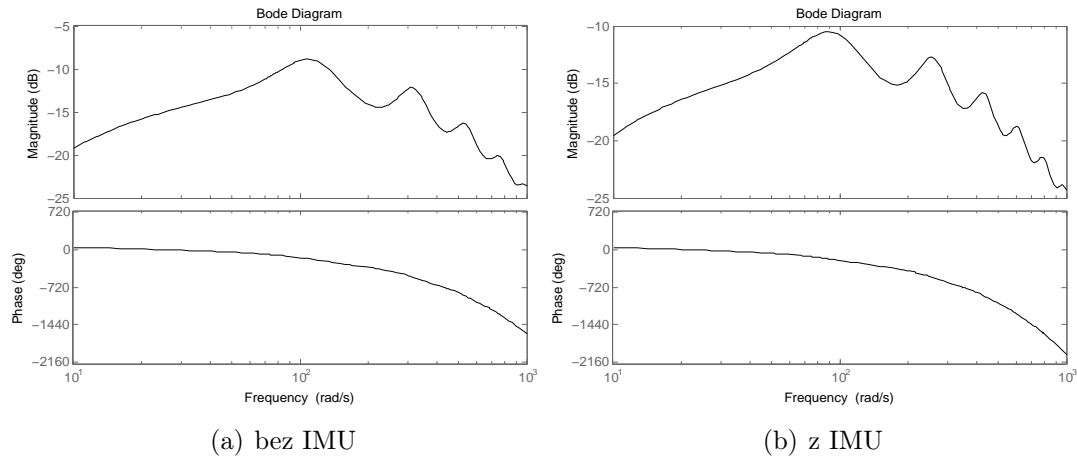
Rysunek 5.21: Wykres prędkości zmierzonej ( $\dot{X}_m$ ) i zadanej ( $\dot{X}_d$ ) z symulacji z dodatkowym tłumieniem

a układ z IMU nie wzbudzał się. Arbitralnie dobranymi parametrami regulatora było  $B_p=500$  i  $M_p=0$ .

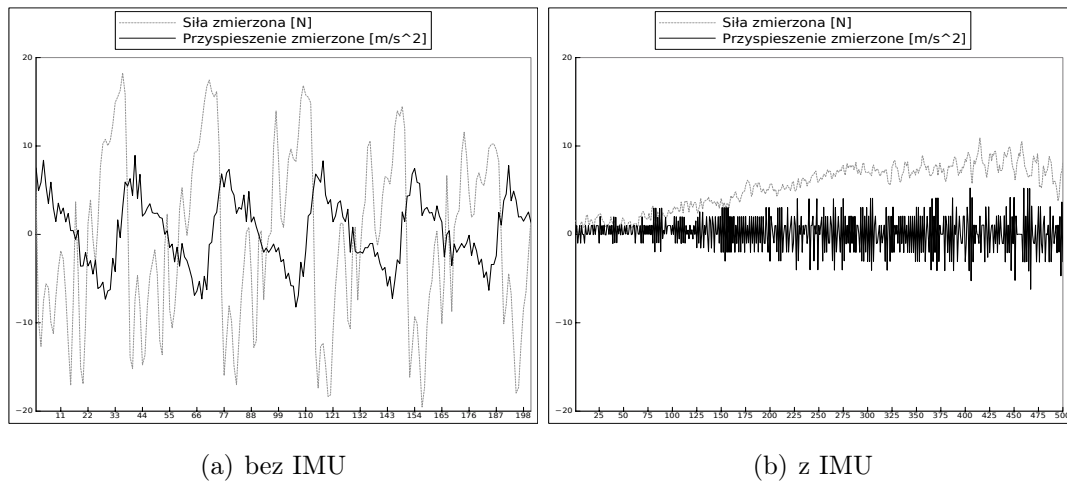
### 5.6.2 Wyniki eksperymentu

Na rysunkach 5.23. i 5.24. znajdują się wykresy wyników eksperymentów bez użycia jednostki inercyjnej, oraz z użyciem jednostki inercyjnej dla eksperymentów z dodatkową masą. Wykresy dla układu bez IMU przedstawiają same drgania, zaś w przypadku układu z IMU proces wzbudzania nie dał wyniku - pomimo naprzemiennego zadawania siły i izolowania końcówki chwytaka od otoczenia obiekt nie wzbudził się.

Wykresy bez IMU są podobne do tych z poprzednich eksperymentów - zauwa-

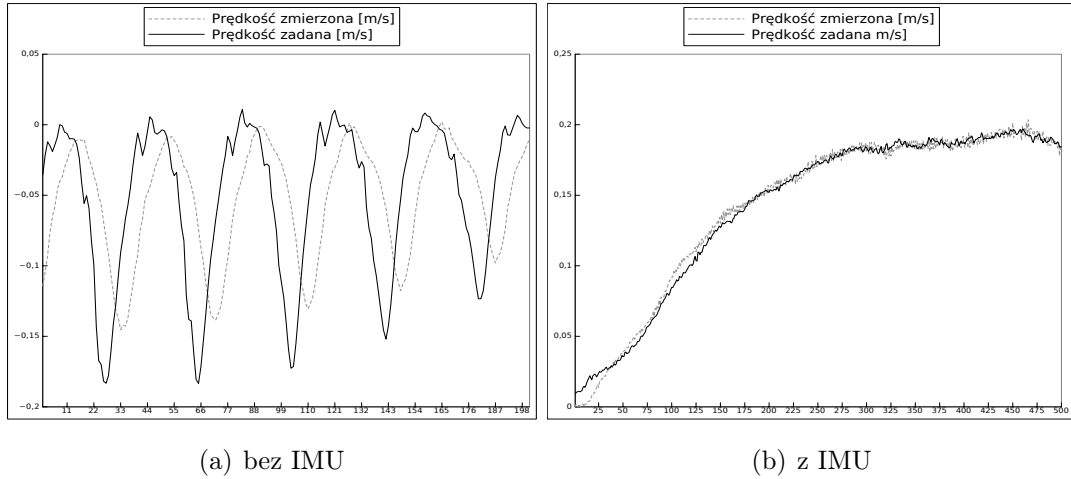


Rysunek 5.22: Wykresy Bodego  $G_Z$  układu z pętlą zamkniętą dla parametrów uzyskanych w eksperymencie 3.



Rysunek 5.23: Wykres siły zmierzonej ( $F_m$ ) i przyspieszenia zmierzonego ( $\ddot{X}_m$ ) dla eksperymentu 5.

żalne są drgania, system się wzbudza. W przypadku systemu z użyciem IMU drgania nie występują. Zadana prędkość jest dobrze naśladowana przez system, nie ma ujemnego wzmocnienia, które występowało w pierwszych eksperymentach. Wygląd wykresu przyspieszenia zmierzonego jest bardzo skwantowany. Spowodowane jest to obliczaniem przyspieszenia jako podwójnie zróżniczkowanej pozycji. Zauważalne jest w eksperymencie duże tłumienie wprowadzane przez regulator, powodujące większy opór robota na zadawaną siłę. Wyniki tego eksperymentu potwierdzają przypuszczenie, że układ z użyciem IMU może posiadać lepsze cechy ruchu niż układ bez IMU.



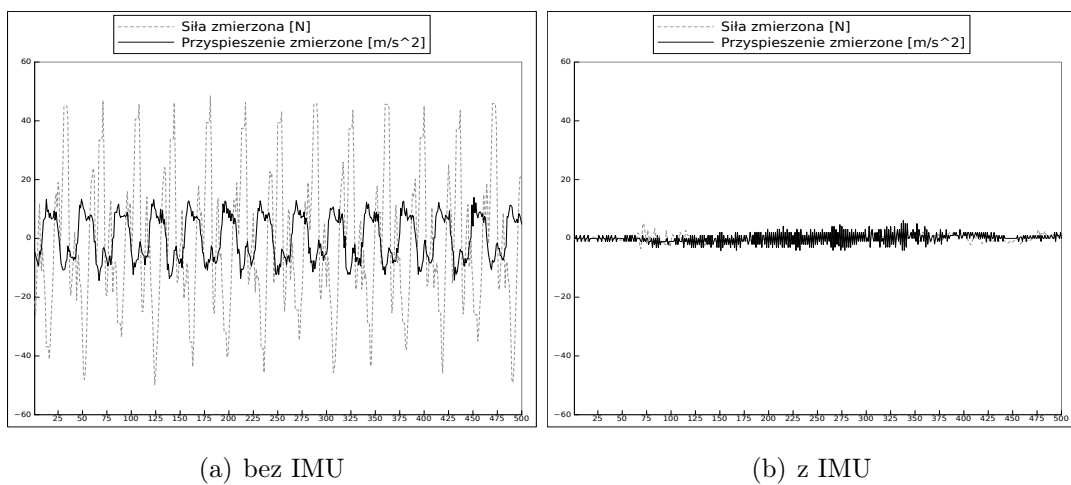
Rysunek 5.24: Wykres prędkości zmierzonej ( $\dot{X}_m$ ) i zadanej ( $\dot{X}_d$ ) dla eksperymentu 5.

## 5.7 Eksperyment 6.

### 5.7.1 Sposób przeprowadzenia eksperymentu

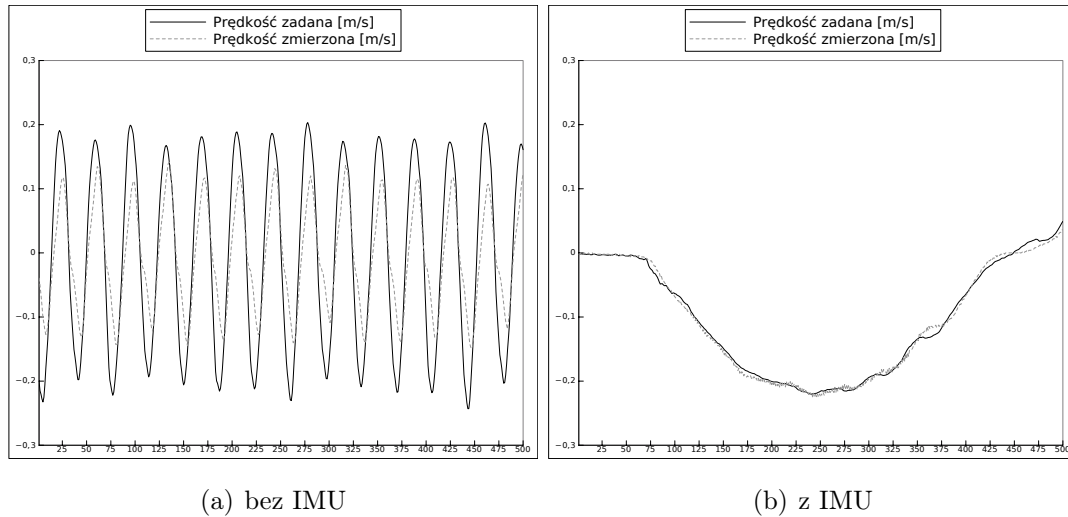
Eksperyment polegał na wykonaniu pomiarów jak w eksperymencie 1. dla takich parametrów regulatora (przez zmianę inercji), aby układ bez IMU się wzbudzał a układ z IMU nie wzbudzał się. Dobranymi parametrami regulatora było  $B_p=100$  i  $M_p=3$ , zostały one dobrane arbitralnie aby zilustrować różnice w działaniu sterownika.

### 5.7.2 Wyniki eksperymentu



Rysunek 5.25: Wykres siły zmierzonej ( $F_m$ ) i przyspieszenia zmierzonego ( $\ddot{X}_m$ ) dla eksperymentu 6.

Na rysunkach 5.25. i 5.26. znajdują się wykresy wyników eksperymentów bez użycia jednostki inercyjnej, oraz z użyciem jednostki inercyjnej dla eksperymentu 6. Wykresy dla układu bez IMU przedstawiają same drgania, zaś w przypadku układu z IMU proces wzbudzenia nie dał wyniku - pomimo naprzemiennego zadawania siły i izolowania końcówki chwytaka od otoczenia obiekt nie wzbudził się.



Rysunek 5.26: Wykres prędkości zmierzonej ( $\dot{X}_m$ ) i zadanej ( $\dot{X}_d$ ) dla eksperymentu 6.

Wykresy bez IMU są podobne do tych z poprzednich eksperymentów - zauważalne są drgania, system się wzbudza. W przypadku systemu z użyciem IMU drgania nie występują. Zadana prędkość jest dobrze naśladowana przez system, nie ma ujemnego wzmocnienia, które występowało w pierwszych eksperymentach. Wygląd wykresu przyspieszenia zmierzonego i siły zmierzonej sugeruje dobre działanie jednostki inercyjnej w tym przypadku, poza początkowymi, względnie dużymi (kilka N) wartościami siły zmierzonej sygnał ma bardzo niskie wartości.

Tablica 5.9: Wartości parametrów dla eksperymentu 1.

Eksperyment	wersja	$B_p$	$M_p$	$M$	$K$	$B$	$Ta$	$Tb$	$Td$
Eksperyment 1	bez IMU	70	0	1.3	0	0	0.00203	0.0296	0.0245
	z IMU	70	0	1.3	0	0	0.00105	0.0136	0.0098
Eksperyment 2	bez IMU	100	0	1.3	0	0	0.00192	0.0256	0.0245
	z IMU	100	0	1.3	0	0	0.00213	0.0116	0.0085
Eksperyment 3	bez IMU	100	0	2.3	0	0	0.00155	0.0336	0.0265
	z IMU	100	0	2.3	0	0	0.00104	0.0206	0.0113
Eksperyment 4	bez IMU	100	0	1.3	0	0.01	0.00107	0.0451	0.0262
	z IMU	100	0	1.3	0	0.01	0.00087	0.0533	0.0332

## 5.8 Podsumowanie eksperymentów

Podsumowanie wartości parametrów użytych i uzyskanych w eksperymentach znajduje się w tablicy 5.9.

Wyniki eksperymentów sugerują, że:

- Siła zmierzona przy użyciu jednostki inercyjnej jest niekiedy większa niż bez użycia jednostki inercyjnej, co sugeruje błąd działania jednostki inercyjnej,
- Model układu nie jest idealny, zamontowanie jednostki inercyjnej zmieniło transmitancję układu (w szczególności modułu  $G_4$ ) - niemożliwe jest dopasowanie wzmocnień tego układu uzyskanych eksperymentalnie do symulacji.





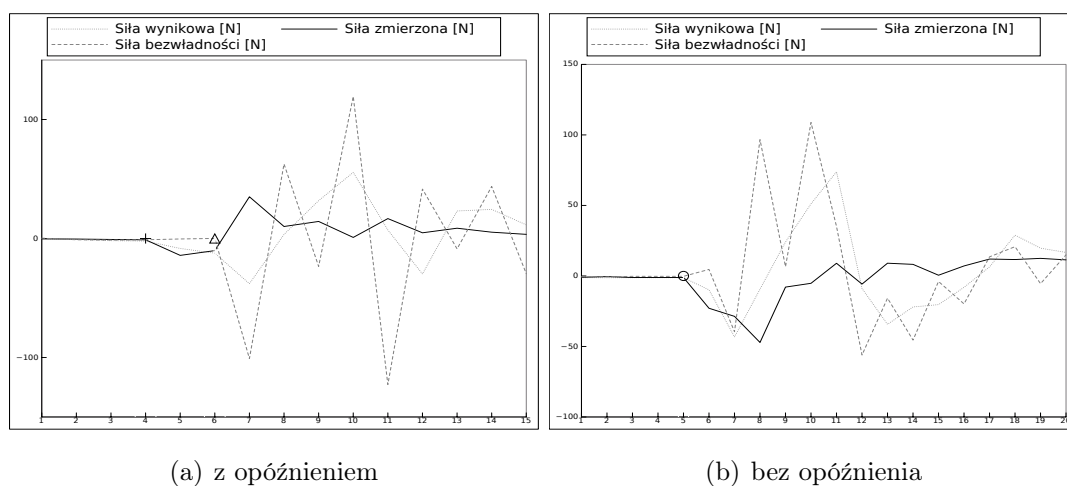
## Rozdział 6

# Weryfikacja pomiarów z użyciem jednostki inercyjnej

W celu ustalenia powodów błędnego działania jednostki inercyjnej zamontowanej na robocie zostały przeprowadzone badania. W rozdziale 6.1. badane było opóźnienie, które pojawia się w systemie. Rozdział 6.2. przedstawia badanie wpływu opóźnienia na korektę siły.

### 6.1 Badanie opóźnienia

W celu oszacowania opóźnienia wprowadzonego przez jednostkę inercyjną przeprowadzono badania mające na celu zadanie skokowego przyspieszenia czujnikom i zbadanie ich zachowania. Wykresy przedstawiające otrzymane wyniki znajdują się na rysunku 6.1.

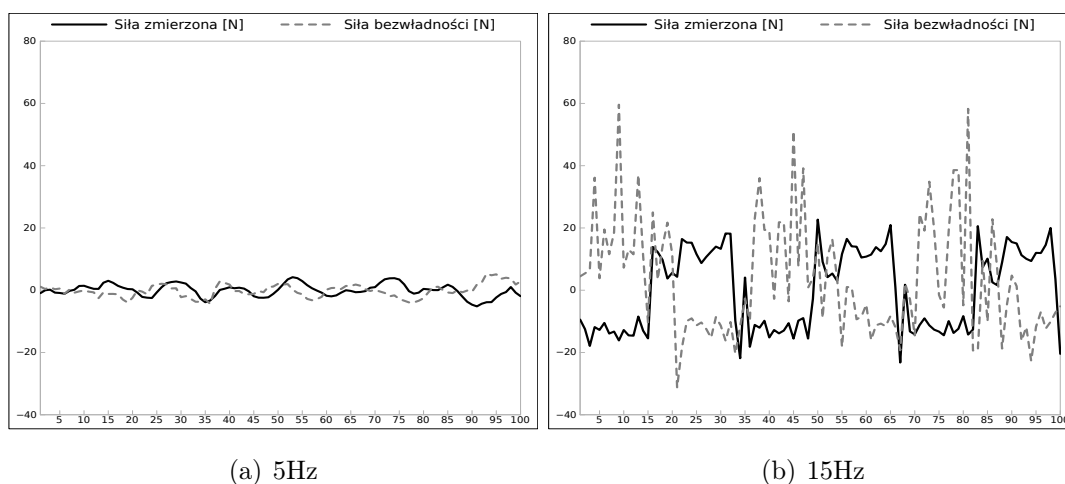


Rysunek 6.1: Wykresy zmierzonej siły, siły bezwładności i wynikowej siły dla eksperymentu z zadaniem przyspieszeniem skokowym.

Na rysunku 6.1(a). znakiem plusa oznaczony został moment odejścia siły zmierzonej od zera, zaś trójkątem moment zmiany wartości siły bezwładności na wartość niezerową. Na tej podstawie widać, że opóźnienie wyniosło 4 ms. Na wykresie 6.1(b). okręgiem został zaznaczony moment, w którym zarówno siła zmierzona przez czujnik siły jak i siła bezwładności zmierzona przez jednostkę inercyjną mają niezerowe odczyty. Kolejne przeprowadzane eksperymenty pokazywały, że opóźnienie pojawia się mniej więcej w co drugim eksperymencie i przyjmuje średnio wartość 4 ms. Zauważalne są też bardzo duże wahania sił - przekraczające amplitudą kilkukrotnie maksymalną siłę zmierzoną przez czujnik siły robota. Może być to spowodowane niezbyt stabilnym umocowaniem jednostki inercyjnej.

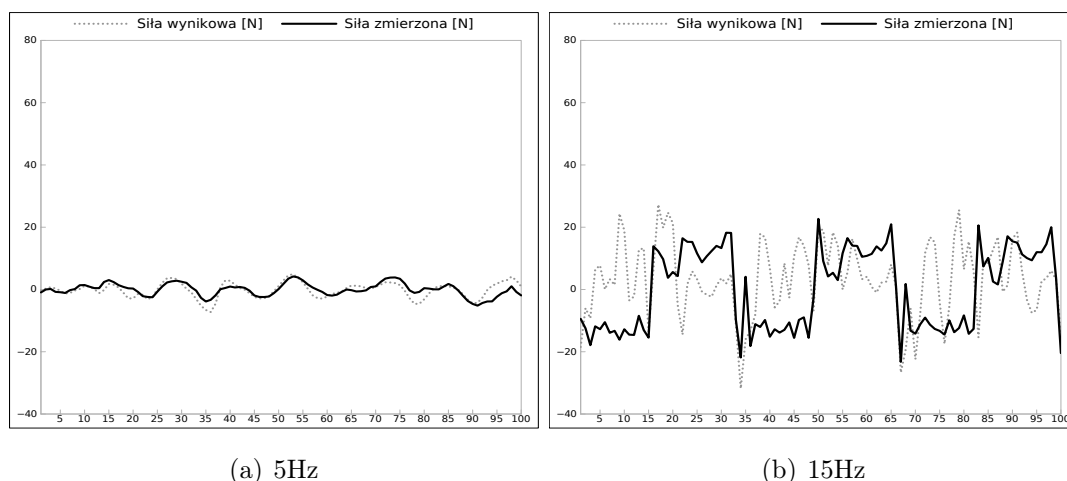
## 6.2 Badanie wpływu opóźnienia na poprawę odczytów siły

Na rysunkach 6.2. i 6.3. znajdują się wykresy przedstawiające siłę zmierzoną, oszacowaną na podstawie wyników z IMU siłę bezwładności oraz siłę wynikową, przeliczoną na podstawie dwóch poprzednich sił. Wykresy zostały uzyskane poprzez zadanie sygnału chirp jako zadaną prędkość dla robota. Zauważalne jest, że zarówno w przypadku sygnału o częstotliwości 5Hz jak i 15Hz maksymalne odczyty siły wynikowej są zdecydowanie większe od odczytów siły zmierzonej. Widoczne jest też przesunięcie fazowe siły bezwładności, która jest opóźniona względem zmierzonej siły.



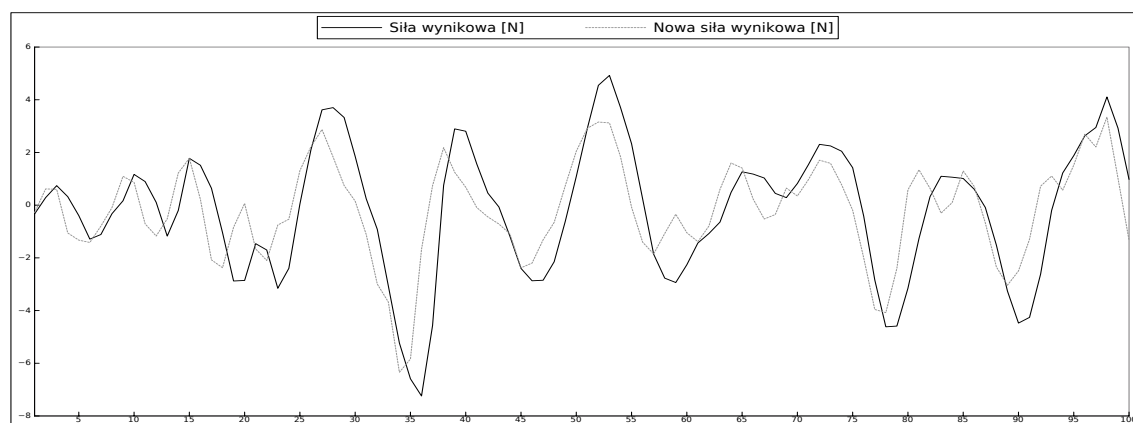
Rysunek 6.2: Wykresy zmierzonej siły i wynikowej siły dla eksperymentu zadanej sinusoidy.

Eksperymenty przeprowadzone poprzez użycie poprawionej siły bezwładności



Rysunek 6.3: Wykresy siły bezwładności i wynikowej siły dla eksperymentu zadanej sinusoidy.

(z odczytem przesuniętym o cztery milisekundy i uśrednionym z dwóch pomiarów) przedstawione są na rysunku 6.4. Na wykresie dla przykładowej częstotliwości 5Hz zmiany pokazują nieznaczną poprawę jakości sygnału zmierzonego.



Rysunek 6.4: Wykresy zmierzonej siły i wynikowej siły dla eksperymentu z zadany przyspieszeniem skokowym po poprawie siły bezwładności.

Wyeliminowanie opóźnień wpływa pozytywnie na przeliczaną siłę. Przypuszczalnym powodem opóźnień jest magistrala USB - nie daje ona gwarancji dostarczenia przesyłanych danych w deterministycznym czasie.



# Rozdział 7

## Identyfikacja członu $G_2$

Badania zostały przeprowadzone w celu weryfikacji poprawności metody określania parametrów układu robotycznego, oraz aby przetestować możliwość stosowania narzędzia System Identification Tool do identyfikacji tych parametrów.

W rozdziale 7.1. omówiony został cel badań, przedstawiona została też postać funkcji  $G_2$  otrzymana w poprzednich eksperymentach i jej odpowiedź na sygnał chirp. W rozdziale 7.2. opisany został sposób przeprowadzenia badań - eksperymentów na robocie, oraz ich analizy przy użyciu narzędzia System Identification Tool. Rozdziały 7.3, 7.4, 7.5. i 7.6. przedstawiają badania przeprowadzone w celu weryfikacji liniowości układu dla różnych amplitud sygnału zadanego i różnych szybkości zmian częstotliwości. W rozdziale 7.7. przedstawiona została próba zastosowania zidentyfikowanego członu  $G_2$  w modelu układu robotycznego i opisane zostały efekty tej próby. Rozdział 7.8. zawiera wnioski płynące z badań opisanych w tym rozdziale.

### 7.1 Cel badań

W tym rozdziale przedstawione będą badania przeprowadzone dla różnych amplitud i prędkości zmian częstotliwości sygnału zadanego. Eksperymenty mają na celu wykonanie przez końcówkę robota ruchów, pozwalających na określenie transmitancji bloku odpowiadającego za przetworzenie prędkości zadanej na prędkość rzeczywistą. W tym celu użyty został generator, zadający prędkość w postaci  $\sin(t^2)$ , dzięki czemu do analizy uzyskane zostały dane w pewnym zakresie częstotliwości. Te dane zostały następnie przeanalizowane przy użyciu narzędzia System Identification Tool.

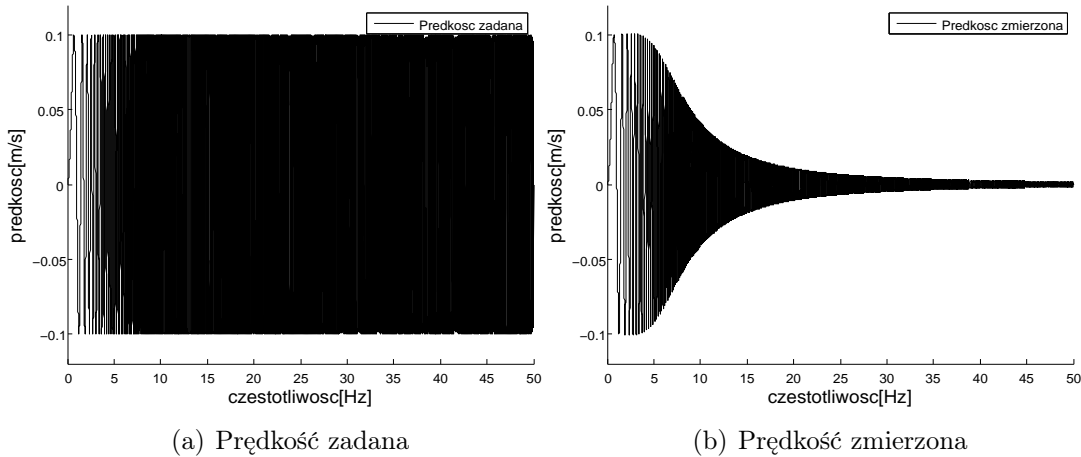
Eksperyment miał na celu pobranie danych do analizy transmitancji przy użyciu narzędzia System Identification Tool. Uznałem, że przy użyciu tego narzędzia można wyznaczyć transmitancję członu  $G_2$  - idealnie sztywnych silników, przekładni

i cięgien. Człon ten jako wejście przyjmuje prędkość zadaną, zaś jego wyjściem jest prędkość zmierzona. W celu jak najdokładniejszego określenia transmitancji układu eksperyment miał stworzyć dane w pewnym zakresie częstotliwości.

Jako bazową wartość funkcji transmitancji dla modelu przyjęto uśrednioną wartość dla wyników bez IMU z eksperymentów przedstawionych w rozdziale 5:

$$\frac{1}{0.0006 * s^2 + 0.032s + 1} \quad (7.1)$$

Wykres prędkości zmierzonej i prędkości zadanej dla tego członu uzyskany z programu SIMULINK przedstawiony jest na wykresie 7.1. Na wykresie widać, że prędkość dla wyższych częstotliwości jest coraz gorzej przenoszona przez układ. Wykresy prędkości zadanej w pozostałych wykresach dla badań w tym rozdziale nie będą pokazywane, jedynie będzie zaznaczana amplituda tej funkcji.



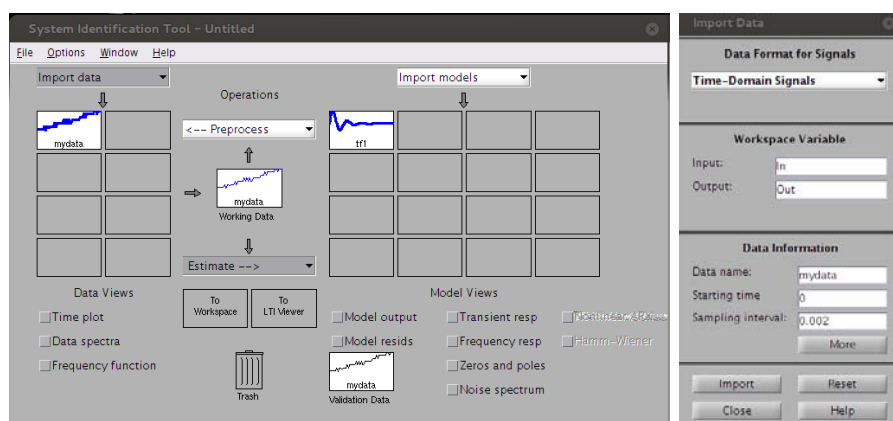
Rysunek 7.1: Prędkość zadana i zmierzona dla symulacji funkcji transmitancji uzyskanej w eksperymentach klasycznych

## 7.2 Sposób przeprowadzenia badań

W celu przeprowadzenia tych eksperymentów stworzyłem sterownik (opisany szerzej w rozdziale 4.2.2), który miał na celu zadanie prędkości w postaci  $\sin(t^2)$ , czyli o częstotliwości zwiększającej się z czasem. Pomiary prędkości zadanej i zmierzonej wykonywane dla tych danych w pewnym zakresie częstotliwości pozwalają na w miarę dokładne wyznaczenie funkcji transmitancji członu  $G_2$ .

Dla danych uzyskanych z eksperymentu na robocie przeprowadzona została analiza funkcji transmitancji przy użyciu System Identification Tool. W głównym oknie programu widocznym na rysunku 7.2(a). wybrana została opcja importowania da-

nych wejściowych i wyjściowych jak na rysunku 7.2(b). Dane te zostały zaimportowane z interwałem próbkowania pomiarów równym 0.002 s.

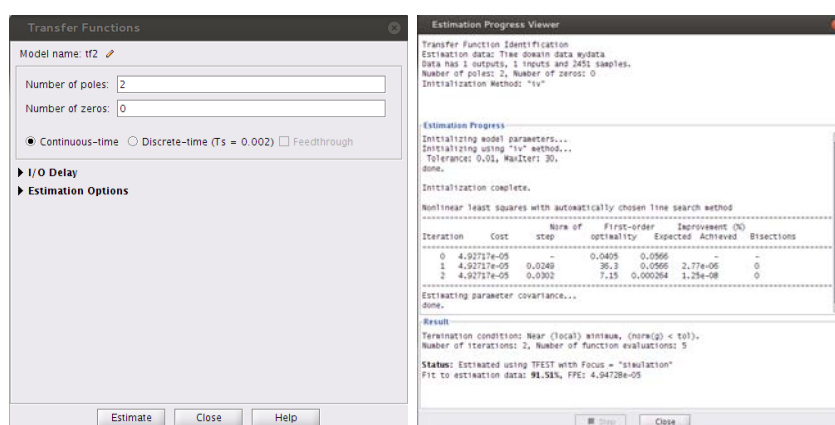


(a) okno główne

(b) okno importowania danych

Rysunek 7.2: Główne okno oraz okno importowania danych System Identification Tool

Następnie wybrana została opcja estymacji parametrów funkcji transmitancji. W oknie wyboru parametrów funkcji transmitancji (rysunek 7.3(a)) zostały wybrane zakładane cechy funkcji  $G_2$  - 4 bieguny i 3 zera, ponieważ taka postać funkcji transmitancji dawała wystarczająco dobre rezultaty dopasowania funkcji wyjściowej do sygnałów. Rysunek 7.3(b). przedstawia okno wynikowe System Identification Tool.



(a) okno wyboru parametrów

(b) okno wynikowe

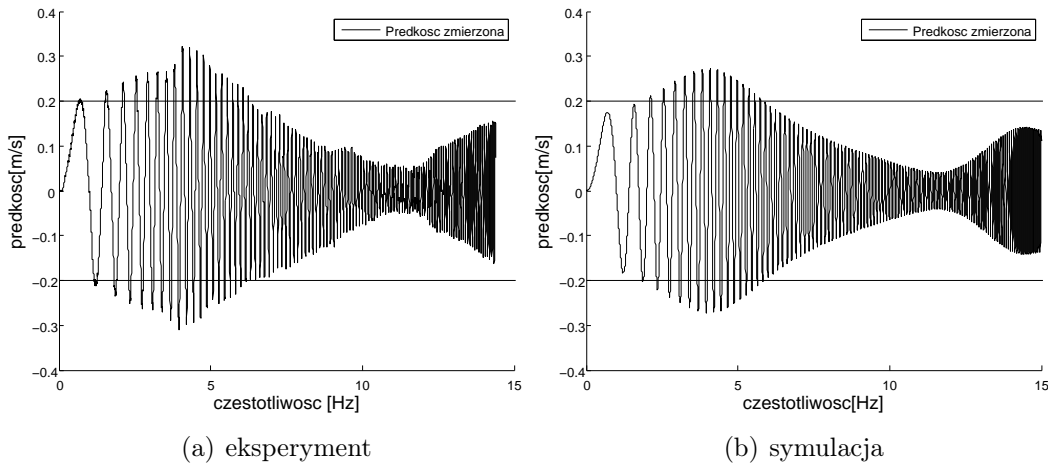
Rysunek 7.3: Okno wyboru parametrów funkcji transmitancji i okno wynikowe

Dla wartości funkcji transmitancji tworzono następnie model w programie SIMULINK, który miał na celu ukazanie odpowiedzi zidentyfikowanego członu na taki

sam zadany sygnał, jak miało to miejsce w eksperymencie (o tej samej amplitudzie i szybkości zmian częstotliwości).

### 7.3 Badanie 1.

Badania zostały przeprowadzone dla amplitudy 0.2 m/s i zmiany częstotliwości sygnału o 1 Hz co 500 kroków (1s). Wyniki eksperymentu na robocie znajdują się na rysunku 7.4(a).



Rysunek 7.4: Wyniki eksperymentu i badań dla amplitudy 0.2m/s i zmiany o 1Hz/1s

Funkcja transmitancji ma postać:

$$\frac{35.9s^3 + 1355s^2 + 208800s + 5065000}{s^4 + 44.36s^3 + 8724s^2 + 209000s + 5917000} \quad (7.2)$$

Zależność wyjścia od wejścia dla tej funkcji (zasymulowana w pakiecie SIMULINK) przedstawiona jest na wykresie 7.4(b).

### 7.4 Badanie 2.

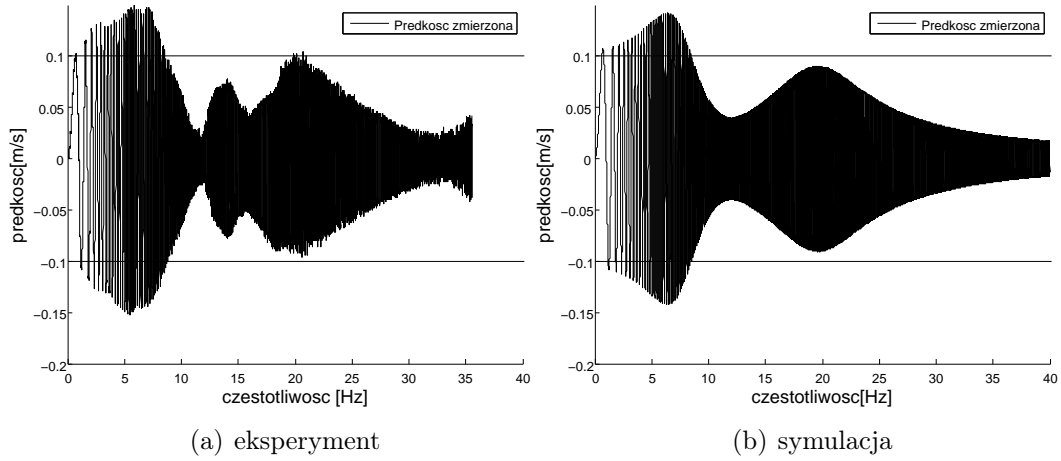
Badania zostały przeprowadzone dla amplitudy 0.1 m/s i zmiany częstotliwości sygnału o 1 Hz co 500 kroków (1s). Wyniki eksperymentu na robocie znajdują się na rysunku 7.5(a). Wykres wynikowy nie jest idealnym odwzorowaniem sygnałów otrzymanych przy użyciu eksperymentów, jednak jest zbliżony do nich.

Funkcja transmitancji ma postać:

$$\frac{20.03s^3 + 7954s^2 + 332500s + 38760000}{s^4 + 85.07s^3 + 19610s^2 + 602300s + 36240000} \quad (7.3)$$

Zależność wyjścia od wejścia dla symulacji tej funkcji przedstawiona jest na wykresie 7.5(b).

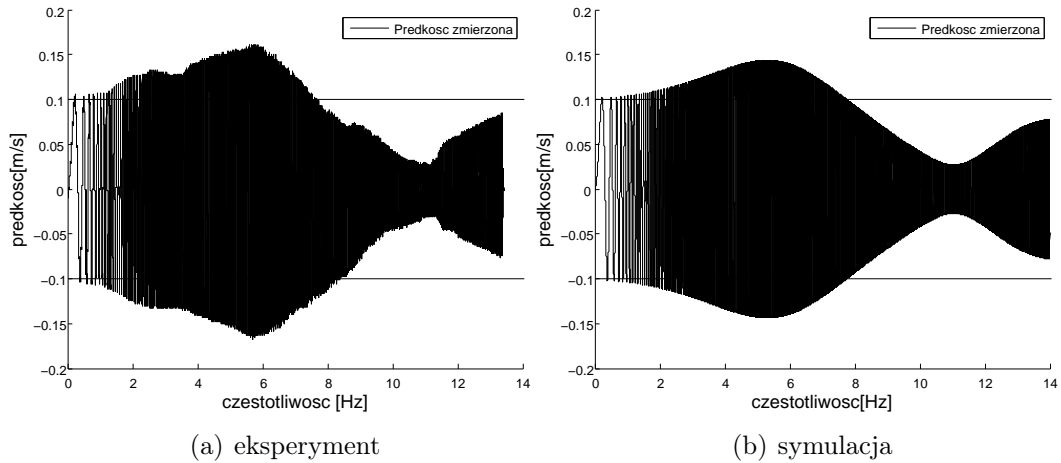




Rysunek 7.5: Wyniki eksperymentu i badań dla amplitudy 0.1m/s i zmiany o 1Hz/1s

## 7.5 Badanie 3.

Badania zostały przeprowadzone dla amplitudy 0.1 m/s i zmiany częstotliwości sygnału o 1 Hz co 5000 kroków (10s). Wyniki eksperymentu na robocie znajdują się na rysunku 7.6(a).



Rysunek 7.6: Wyniki eksperymentu i badań dla amplitudy 0.1m/s i zmiany o 0.1Hz/1s

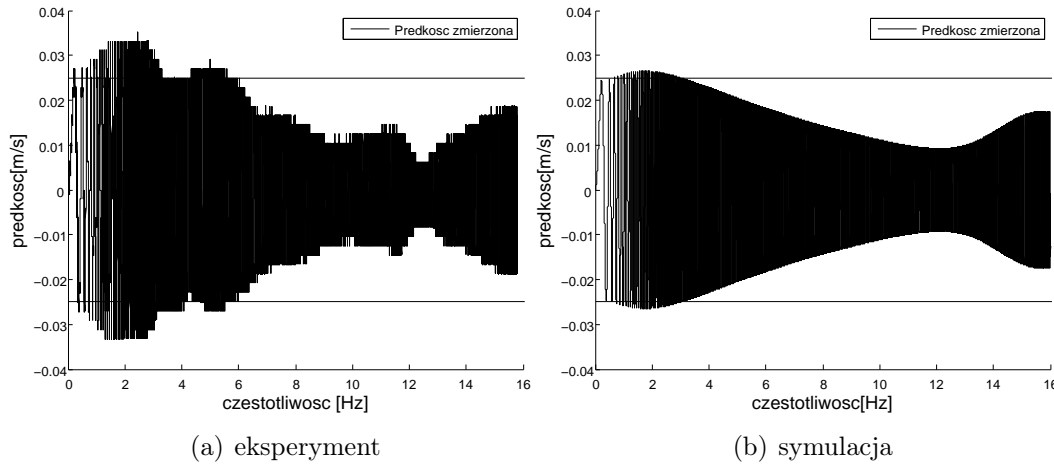
Funkcja transmitancji ma postać:

$$\frac{51.14s^3 + 2899s^2 + 274200s + 11510000}{s^4 + 66.79s^3 + 9514s^2 + 296500s + 11290000} \quad (7.4)$$

Zależność wyjścia od wejścia dla symulacji tej funkcji przedstawiona jest na wykresie 7.6(b).

## 7.6 Badanie 4.

Badania zostały przeprowadzone dla amplitudy 0.025 m/s i zmiany częstotliwości sygnału o 1 Hz co 5000 kroków (10s). Wyniki eksperymentu na robocie znajdują się na rysunku 7.7(a).



Rysunek 7.7: Wyniki eksperymentu i badań dla amplitudy 0.025m/s i zmiany o 0.1Hz/1s

Funkcja transmitancji ma postać:

$$\frac{48.44s^3 + 1702s^2 + 343400s + 2683000}{s^4 + 65.25s^3 + 10290s^2 + 364700s + 2763000} \quad (7.5)$$

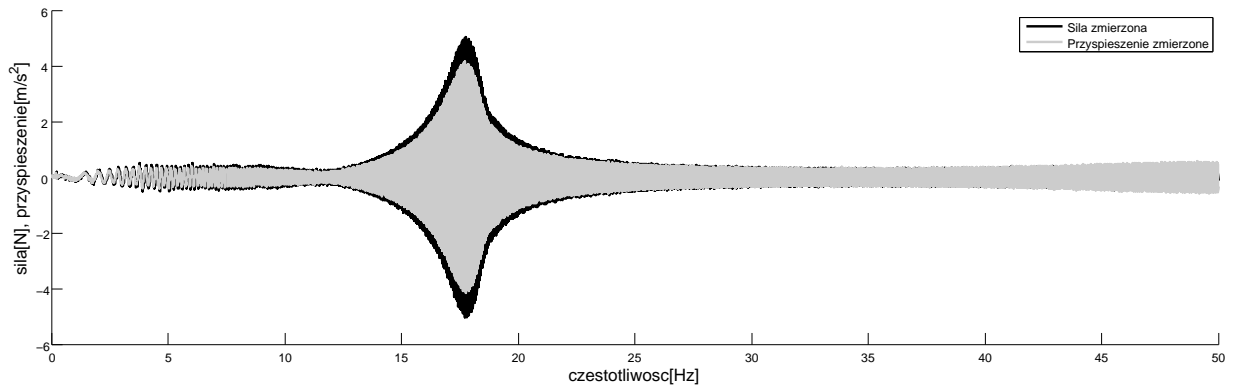
Zależność wyjścia od wejścia dla symulacji tej funkcji przedstawiona jest na wykresie 7.7(b).

## 7.7 Weryfikacja działania zidentyfikowanego członu $G_2$ w układzie

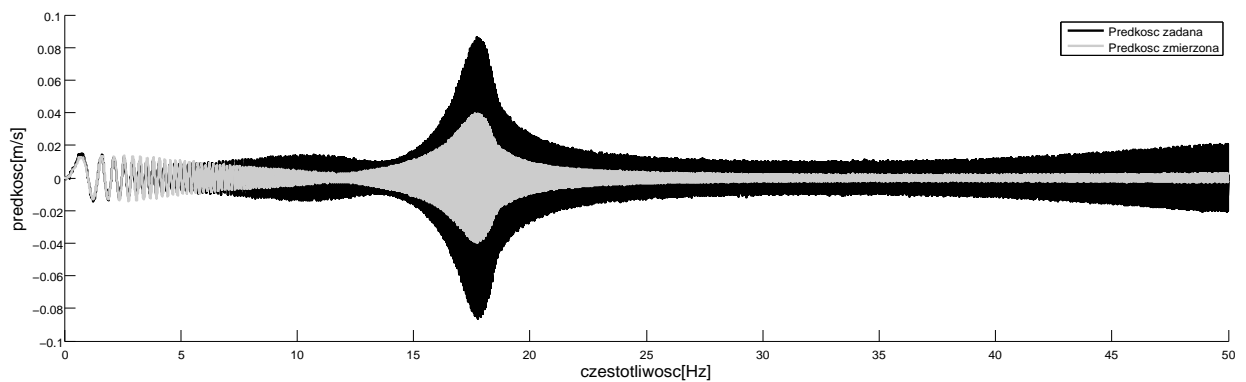
W celu weryfikacji uzyskanych parametrów przetestowano je w modelu robota jako człon  $G_2$ . Otrzymane wykresy siły, przyspieszenia i prędkości przedstawione są na rysunkach 7.8. i 7.9. Dla wykresu siły i przyspieszenia zmierzonego można zauważyć, że od częstotliwości równej około 30Hz wykresy się pokrywają.

## 7.8 Wnioski z badań

Identyfikacja członu  $G_2$  przy użyciu narzędzia System Identification Tool jest poprawna. Wykresy sygnałów uzyskanych z przesymulowania otrzymanych funkcji są bardzo podobne do wykresów otrzymanych w eksperymentach. Częstotliwość



Rysunek 7.8: Siła zmierzona i przyspieszenie zmierzone dla układu ze zidentyfikowanym członem  $G_2$  w układzie z pętlą zamkniętą



Rysunek 7.9: Prędkość zadana i zmierzona dla układu ze zidentyfikowanym członem  $G_2$  w układzie z pętlą zamkniętą

rezonansowa odczytana z wykresu symulacji z pętlą zamkniętą wynosi ok. 18Hz, wzmocnienie członu  $G_2$  około -6, zaś wzmocnienie członu  $G_{3I4}$  około 2, co sugeruje, że symulacja układu z pętlą zamkniętą nie jest poprawna mimo identyfikacji członu  $G_2$ . Jest to prawdopodobnie spowodowane uproszczeniem, jakim jest ten model w stosunku do rzeczywistego układu. Przybliżenie transmitancji  $G_2$  przy użyciu członu dwuinercyjnego zakładało przeniesienie opóźnień płynących z cięgien i przekładni do kolejnych członów. Dokładna identyfikacja członu  $G_2$  przeprowadzona przy użyciu System Identification Tool powinna nieść ze sobą zmiany też w innych członach układu. Człon nie jest dokładnie liniowy, jednak zauważalne jest niezależnie od amplitudy sygnału zadanego wzmocnienie powyżej jedności dla niskich częstotliwości, znaczne zmniejszenie wzmocnienia dla około 11-12 Hz i następnie wzrost wzmocnienia.



# Rozdział 8

## Wnioski

W ramach pracy magisterskiej przebadany został model układu robotycznego. Jego parametry zostały wyznaczone przy użyciu nowych metod (analiza widma częstotliwościowego sygnału, wykorzystanie System Identification Tool w programie MATLAB) poprzez analizę przeprowadzonych eksperymentów.

Wykorzystane narzędzia, zarówno MATLAB i jego pakiety jak i LabVIEW są odpowiednie dla przetwarzania sygnałów, obliczeń i symulacji.

Wykorzystywany model robota, stworzony w programie SIMULINK wydaje się być dość dokładny dla pewnego zakresu częstotliwości, niestety nie można go ekstrapolować na wyższe częstotliwości.

Pomiary z użyciem jednostki inercyjnej są obarczone pewnym opóźnieniem, które sprawia, że dla wyższych częstotliwości pomiary są niedokładne. Może to być spowodowane zastosowaniem protokołu USB do transmisji danych.

Sposób użycia jednostki inercyjnej w robotach spowodował przesunięcie szczytu rezonansowego do wyższych częstotliwości. Jednak w przypadku wyeliminowania drgań jednostki inercyjnej i opóźnienia sygnałów odczytywanych z niej można się spodziewać dalszej poprawy działania systemu robotycznego.



# Bibliografia

- [1] T. Winiarski and K. Banachowicz. System akwizycji skorygowanej siły uogólnionej kontaktu robota manipulacyjnego z otoczeniem. *Pomiary Automatyka Robotyka*, (2):390–394, 2013.
- [2] T. Winiarski and A. Woźniak. Indirect force control development procedure. *Robotica (Cambridge Journals Online)*, 1(1):1–14, 16 August 2012.
- [3] T. Kroger, D. Kubus, and F. M. Wahl. Force and acceleration sensor fusion for compliant manipulation control in 6 degrees of freedom. *Advanced Robotics*, 21(14):1603–1616, 2007.
- [4] G. Bechciński. Konstrukcja i działanie oraz obsługa robota przemysłowego IRp–6 <http://ioitbm.p.lodz.pl/Dydaktyka/Obrabiarki/R/R-4.pdf>, Łódź, 2011.
- [5] Cezary Zieliński, Wojciech Szynkiewicz, Tomasz Winiarski, and Tomasz Kornuta. MRROC++ Based System Description. Technical report, IAIIS, Warszawa, Czerwiec 2007.
- [6] B. Mrozek, Z. Mrozek, and J. Mościński. *Matlab: uniwersalne środowisko do obliczeń naukowo-technicznych*. Wydaw. PLJ, 1996.
- [7] Inc. The Mathworks. Matlab documentation <http://www.mathworks.com/help/matlab>, 2013.
- [8] J. Brzózka and L. Dorobczyński. *MATLAB: środowisko obliczeń naukowo-technicznych*. Wydawnictwo Naukowe PWN, 2008.
- [9] National Instruments Corporation. Online ni labview help resources <http://www.ni.com/gettingstarted/labviewbasics/onlinehelp.htm>, 2012.