

Politechnika Wrocławska

Wydział Elektroniki, Fotoniki i Mikrosystemów

KIERUNEK: Automatyka i Robotyka (AIR)

PRACA DYPLOMOWA INŻYNIERSKA

TYTUŁ PRACY:
Identyfikacja parametrów manipulatora z
wykorzystaniem sterowania predykcyjnego
opartego na modelu

AUTOR:
Jan Rybarz

PROMOTOR:
dr inż. Wojciech Domski

Pracę tę dedykuję mojemu tacie, Markowi Rybarzowi, który swoim przykładem i wsparciem zainspirował mnie do ciągłego rozwoju i zdobywania wiedzy. Jego osiągnięcie w postaci uzyskania tytułu doktora inżyniera jest dla mnie nieustanną motywacją.

Serdeczne podziękowania kieruję także do mojego promotora, doktora inżyniera Wojciecha Domskiego, za jego nieocenione wsparcie, wskazówki i cierpliwość podczas realizacji tej pracy.

STRESZCZENIE

Niniejsza praca inżynierska dotyczy identyfikacji parametrów manipulatora typu 3R z wykorzystaniem sterowania predykcyjnego opartego na modelu. W pracy przedstawiono model matematyczny manipulatora, jego dynamikę oraz teoretyczne podstawy sterowania predykcyjnego. Zaimplementowano algorytm MPC w języku Python, stosując metodę różnicowej ewolucji do optymalizacji parametrów modelu, takich jak masy i długości ogniw. Przeprowadzono symulacje o różnych wartościach parametrów, analizując wpływ horyzontu predykcji, czasu próbkowania oraz ograniczeń na dokładność identyfikacji. Wyniki wykazały, że odpowiedni dobór parametrów algorytmu, pozwala na uzyskanie wysokiej precyzji estymacji. Szczególną rolę odegrało uwzględnienie ograniczeń fizycznych, co znacznie poprawiło zgodność wyników z rzeczywistymi wartościami. Podkreślono również znaczenie funkcji kosztu skoncentrowanej na błędach trajektorii. Ostatecznie, przeprowadzone badania potwierdziły skuteczność metody MPC w identyfikacji parametrów manipulatorów i otworzyły nowe kierunki dalszych badań, m.in. integrację algorytmów uczenia maszynowego czy rozszerzenie modelu o dodatkowe czynniki dynamiczne.

SUMMARY

This engineering thesis focuses on the identification of parameters of a 3R manipulator using model predictive control. The study presents the mathematical model of the manipulator, its dynamics, and the theoretical foundations of predictive control. The MPC algorithm was implemented in Python, employing the differential evolution method to optimize model parameters such as segment masses and lengths. Simulations with various parameter values were conducted, analyzing the impact of the prediction horizon, sampling time, and constraints on identification accuracy. The results demonstrated that the appropriate selection of algorithm parameters enables high estimation precision. The inclusion of physical constraints played a significant role, substantially improving the consistency of the results with actual values. The importance of a cost function focused on trajectory errors was also emphasized. Ultimately, the research confirmed the effectiveness of the MPC method for parameter identification in manipulators and opened new avenues for further studies, including the integration of machine learning algorithms and the extension of the model to incorporate additional dynamic factors.

Słowa kluczowe: sterowanie predykcyjne, identyfikacja parametrów, manipulator 3R, optymalizacja numeryczna, dynamika manipulatora

Keywords: predictive control, parameter identification, 3R manipulator, numerical optimization, manipulator dynamics

Spis treści

1	Wstęp	3
1.1	Teza	4
1.2	Podział pracy	5
2	Obiekt sterowania	7
2.1	Kinematyka manipulatora	7
2.2	Algorytm odsprzęgania wejściowo-wyjściowy	9
3	Sterowanie predykcyjne	11
3.1	Opis metody MPC	11
3.2	Wymagania MPC wobec optymalizatora	12
3.3	Formułowanie problemu identyfikacji parametrów	13
3.4	Horyzont predykcji	14
3.5	Dobór współczynników funkcji kosztu i ograniczeń parametrów	15
4	Implementacja algorytmu MPC	17
4.1	Funkcja kosztu	17
4.2	Predykcja trajektorii	18
4.3	Optymalizacja parametrów	19
5	Wyniki symulacji i analiza wyników	21
5.1	Wpływ ograniczeń na dokładność identyfikacji parametrów	21
5.2	Dobór współczynników funkcji kosztu	22
5.3	Wpływ ilości kroków predykcji	23
5.4	Wpływ czasu próbkowania	24
5.5	Wpływ dokładności optymalizatora	24
5.6	Wpływ ilości badanych parametrów na dokładność	25
6	Podsumowanie	27
	Bibilografia	27

Rozdział 1

Wstęp

Rozwój automatyki i robotyki jest ściśle związany z potrzebami przemysłu, który od XIX wieku, za sprawą rewolucji przemysłowej, dążył do zwiększenia wydajności i automatyzacji procesów produkcyjnych. Początkowo maszyny, napędzane parą, wspomagały mechanicznie pracę człowieka, jednak dopiero wynalezienie silnika elektrycznego, a później rozwój technologii komputerowych, stworzyły warunki do zaprojektowania nowoczesnych robotów. Rozwój manipulatorów robotycznych można prześledzić na podstawie zaawansowanych modeli matematycznych, które zostały szczegółowo opisane w [[5]]. Wykorzystanie energii elektrycznej i rozwój elektroniki umożliwiły budowę zaawansowanych układów sterowania, które pozwalały na automatyzację zadań wymagających precyzji i powtarzalności, takich jak montaż, obróbka materiałów czy spawanie.

W latach 50. i 60. XX wieku roboty przemysłowe zaczęły pojawiać się na liniach produkcyjnych, szczególnie w przemyśle motoryzacyjnym, gdzie wykorzystywano je do prac montażowych. Firmy takie jak General Motors wdrażały pierwsze manipulatory, które charakteryzowały się prostą budową i były ograniczone do wykonywania powtarzalnych czynności. Jednak rozwój technologii, w tym mikroprocesorów oraz systemów sterowania cyfrowego, pozwolił na wdrażanie bardziej zaawansowanych algorytmów sterowania, które znacznie zwiększyły możliwości robotów. Pojawiły się roboty o większej elastyczności, zdolne do wykonywania różnorodnych operacji, co przyczyniło się do wzrostu zainteresowania badaniami nad nowymi metodami sterowania.

Sterowanie w robotyce ewoluowało od prostych układów sprzężenia zwrotnego, przez sterowanie PID, aż po nowoczesne metody adaptacyjne i optymalne. Sterowanie adaptacyjne pozwala na dynamiczne dostosowywanie parametrów regulatora do zmieniających się warunków pracy robota, co jest szczególnie ważne w środowiskach, gdzie występują nieprzewidziane zakłócenia lub zmiany charakterystyki obiektu sterowania. Jednak metody te mają swoje ograniczenia, szczególnie gdy w grę wchodzi złożoność dynamiczna systemów wieloczołowych, takich jak manipulatory robotyczne. Dlatego też coraz większą popularność zyskuje sterowanie predykcyjne oparte na modelu *Model Predictive Control*, które umożliwia przewidywanie przyszłego zachowania układu i optymalizację trajektorii ruchu w krótkich odstępach czasu.

Metoda MPC wywodzi się z teorii optymalnego sterowania i bazuje na aktualizacji trajektorii w czasie rzeczywistym poprzez minimalizację funkcji kosztu, która uwzględnia zarówno błędy pozycjonowania, jak i wymagania energetyczne. MPC pozwala na uwzględnienie ograniczeń fizycznych, takich jak maksymalne prędkości i przyspieszenia, co czyni ją szczególnie przydatną w aplikacjach wymagających wysokiej precyzji. Przewidując przyszłe stany układu na podstawie aktualnych informacji, algorytm MPC jest w stanie szybko reagować na zmiany warunków operacyjnych, co znajduje zastosowanie w wielu

dziedzinach przemysłu – od procesów chemicznych, przez zarządzanie sieciami energetycznymi, aż po zaawansowane systemy autonomiczne, takie jak pojazdy bezzałogowe i roboty medyczne.

W kontekście manipulacji robotycznej, MPC jest szczególnie cenione za swoją zdolność do precyzyjnego sterowania trajektorią efektora końcowego. W zadaniach takich jak montaż komponentów na liniach produkcyjnych, precyzyjna kontrola trajektorii jest kluczowa, aby uniknąć błędów, które mogą prowadzić do uszkodzeń lub wadliwych produktów. Również w obszarze chirurgii robotycznej, gdzie manipulatory muszą operować z bardzo wysoką dokładnością, zastosowanie MPC może znacząco poprawić bezpieczeństwo i skuteczność procedur medycznych. Praktyczne wykorzystanie MPC wymaga jednak dokładnej identyfikacji parametrów manipulatora, takich jak masy segmentów, długości ramion, momenty bezwładności oraz współczynniki tarcia. Identyfikacja tych parametrów jest kluczowa, ponieważ wpływają one bezpośrednio na dynamikę układu i dokładność sterowania.

Identyfikacja parametrów dynamicznych manipulatora jest zadaniem wymagającym, zwłaszcza gdy bierze się pod uwagę wpływ zmiennych warunków pracy i niepewności modelu. W praktyce, rzeczywiste manipulatory różnią się od idealizowanych modeli matematycznych, a błędy w oszacowaniach parametrów mogą prowadzić do obniżenia dokładności sterowania. Dlatego stosowanie zaawansowanych metod, takich jak optymalizacja numeryczna, które mogą kompensować te niepewności, jest niezbędne. Wykorzystanie MPC w identyfikacji parametrów pozwala na jednoczesne uwzględnienie błędów pomiarowych oraz ograniczeń fizycznych manipulatora, co przekłada się na bardziej precyzyjne oszacowanie jego parametrów dynamicznych. Ponadto, uwzględnianie wielowymiarowych trajektorii ruchu oraz dynamicznych zmian sił zewnętrznych pozwala na lepsze dostosowanie modelu do rzeczywistych warunków pracy.

Współczesne badania nad robotyką skupiają się nie tylko na optymalizacji istniejących metod sterowania, ale również na integracji sztucznej inteligencji z algorytmami sterowania, co otwiera nowe możliwości adaptacji i uczenia się przez systemy robotyczne. Przykładem może być zastosowanie metod uczenia maszynowego do optymalizacji parametrów algorytmu MPC, co pozwala na automatyczne dostosowywanie strategii sterowania w zależności od bieżących potrzeb i warunków środowiskowych. Tego rodzaju podejście jest szczególnie obiecujące w kontekście robotyki medycznej, eksploracji kosmosu oraz pojazdów autonomicznych, gdzie adaptacyjność systemów może znacząco zwiększyć ich efektywność i bezpieczeństwo.

Celem niniejszej pracy jest zastosowanie metody MPC do identyfikacji parametrów manipulatora typu 3R, czyli manipulatora trójprzegubowego. W pracy tej zostanie przedstawiony model matematyczny manipulatora oraz algorytm MPC, które pozwolą na symulację działania manipulatora oraz porównanie wyników z rzeczywistymi wartościami. Szczególny nacisk zostanie położony na ocenę dokładności identyfikacji parametrów dynamicznych, a także na wpływ horyzontu predykcji i czasu próbkowania na rezultaty. W ramach badań przeprowadzone zostaną testy symulacyjne w różnych scenariuszach operacyjnych, mających na celu ocenę skuteczności proponowanej metody w różnych warunkach.

1.1 Teza

Identyfikacja parametrów manipulatora 3R za pomocą sterowania predykcyjnego opartego na modelu (MPC) pozwala na uzyskanie wysokiej dokładności symulacji, zbliżonej do rze-

czywistych wartości dynamicznych manipulatora, przy odpowiednim doborze horyzontu predykcji i czasu próbkowania.

1.2 Podział pracy

W rozdziale drugim przedstawiono szczegółowy opis obiektu sterowania, czyli manipulatora 3R. Omówiono kinematykę i dynamikę, wskazując, jak poszczególne parametry wpływają na zachowanie systemu. Zajęto się także metodami sterowania oraz technikami dekorelacji, które upraszczają sterowanie.

W rozdziale trzecim opisano teoretyczne podstawy i implementację sterowania predykcyjnego dla manipulatora 3R. Szczegółowo wyjaśniono zasadę działania MPC, przedstawiono funkcję kosztu oraz opisano zastosowane metody optymalizacji i sposoby radzenia sobie z ograniczeniami układu.

Rozdział czwarty zawiera implementację algorytmu MPC w języku Python oraz jego integrację z modelem manipulatora. Przedstawiono analizę parametrów kalibracyjnych i omówiono wpływ horyzontu predykcji na działanie algorytmu.

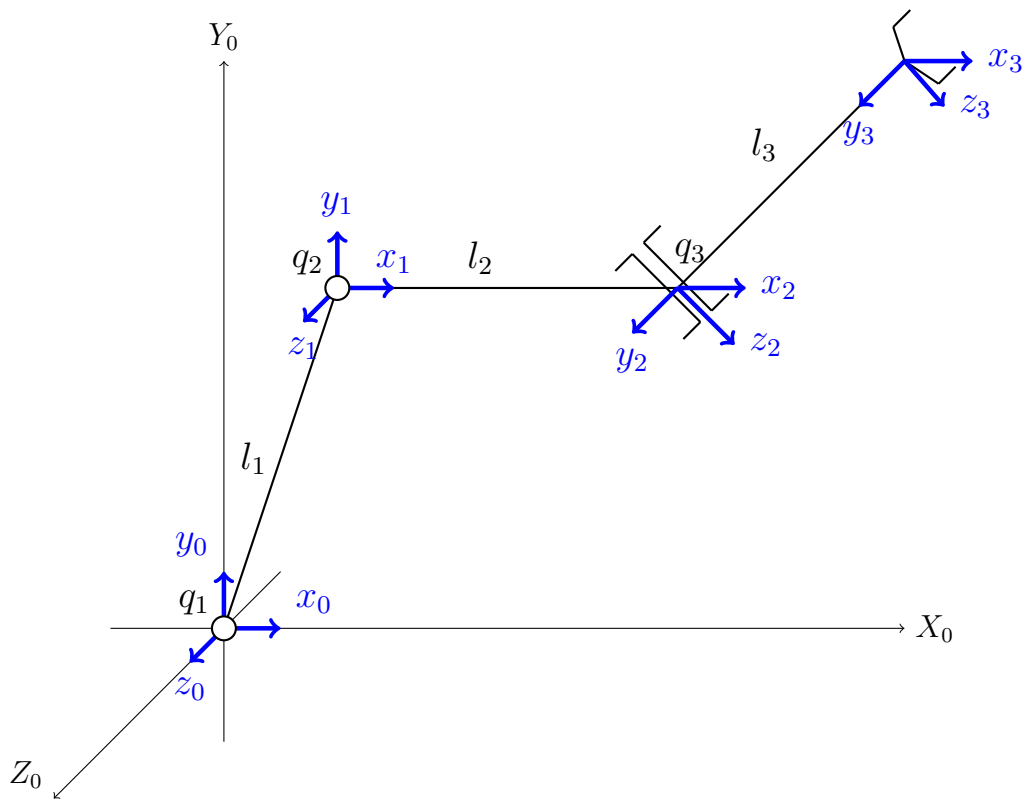
W rozdziale piątym zaprezentowano wyniki symulacji dla różnych konfiguracji parametrów, analizując dokładność identyfikacji parametrów dynamicznych. Przedstawiono również wpływ poszczególnych parametrów na błędy pozycjonowania.

W rozdziale szóstym sformułowano wnioski na temat skuteczności zastosowanej metody MPC w identyfikacji parametrów manipulatora 3R. Omówiono także potencjalne kierunki dalszych badań, które mogłyby udoskonalić metodę.

Rozdział 2

Obiekt sterowania

W rozdziale przedstawiono szczegółowy opis obiektu sterowania, którym jest manipulator typu 3R (trójprzegubowy manipulator obrotowy) działający w przestrzeni trójwymiarowej. Omówiono kinematykę prostą, dynamikę manipulatora oraz metody sterowania, w tym algorytm rozprzęgania wejścia-wyjścia, który umożliwia uproszczenie procesu sterowania.



Rysunek 2.1 Schemat manipulatora typu 3R

2.1 Kinematyka manipulatora

Kinematyka manipulatora opisuje zależności geometryczne między położeniem i orientacją efektora końcowego a kątami przegubów manipulatora. Dzieli się ona na kinematykę prostą oraz odwrotną. W kontekście tego opracowania skupimy się na kinematyce prostej.

Kinematyka prosta

Kinematyka prosta pozwala obliczyć położenie efektora na podstawie znanych kątów przegubów. Dla manipulatora 3R działającego w przestrzeni trójwymiarowej, położenie efektora (x, y, z) w układzie współrzędnych bazowych jest opisane równaniami (2.1).

$$k(q) = \begin{pmatrix} x \\ y \\ z \end{pmatrix} = \begin{pmatrix} l_1 \cos(q_1) + \cos(q_1 + q_2)(l_2 + l_3 \cos(q_3)) \\ l_1 \sin(q_1) + \sin(q_1 + q_2)(l_2 + l_3 \cos(q_3)) \\ l_3 \sin(q_3) \end{pmatrix} \quad (2.1)$$

gdzie:

- l_1, l_2, l_3 – długości pierwszego, drugiego i trzeciego ogniwa manipulatora,
- q_1, q_2, q_3 – kąty w przegubach.

Pierwszy przegub q_1 obraca ramię wokół osi Z , co powoduje ruch w płaszczyźnie XY . Drugi przegub q_2 umożliwia podnoszenie i opuszczanie ramienia w płaszczyźnie pionowej. Trzeci przegub q_3 wpływa na dodatkowe podnoszenie lub opuszczanie efektora, zmieniając położenie w osi Z .

Notacja Denavita-Hartenberga

Do analizy kinematyki manipulatora zastosowano notację Denavita-Hartenberga, która umożliwia jednoznaczne opisanie konfiguracji przestrzennej manipulatora poprzez zestaw parametrów dla każdego ogniwa. Parametry DH dla rozważanego trójprzegubowego manipulatora obrotowego 3R przedstawia tabela 2.1.

i	a_i	α_i	d_i	θ_i
1	l_1	0	0	q_1
2	l_2	$\pi/2$	0	q_2
3	l_3	0	0	q_3

Tabela 2.1 Tabela parametrów Denavita-Hartenberga

gdzie:

- a_i – odległość mierzona wokół osi X_i od Z_{i-1} do Z_i
- α_i – kąt mierzony wokół osi X_i od Z_{i-1} do Z_i
- d_i – przesunięcie liniowe mierzone wzdłuż osi Z_{i-1} od X_{i-1} do X_i
- θ_i – kąt mierzony wokół osi Z_{i-1} od X_{i-1} do X_i

Trajektorie efektora

W celu sterowania manipulatorem konieczne jest zdefiniowanie trajektorii, po której efektor ma się poruszać. W implementacji wykorzystano generator trajektorii, który tworzy różne kształty ruchu w przestrzeni trójwymiarowej. Przykładem jest trajektoria eliptyczna opisana równaniami parametrycznymi (2.2).

$$q_d(t) = \begin{pmatrix} x_d(t) \\ y_d(t) \\ z_d(t) \end{pmatrix} = \begin{pmatrix} A \cos(\omega t) + x_0 \\ B \sin(\omega t) + y_0 \\ C \sin(\omega t) + z_0 \end{pmatrix} \quad (2.2)$$

gdzie:

- A, B, C – amplitudy ruchu w osiach x, y, z ,
- ω – częstość kątowna,
- x_0, y_0, z_0 – przesunięcia początkowe.

2.2 Algorytm odsprężania wejściowo-wyjściowy

Odsprężanie wejścia-wyjścia to metoda umożliwiająca sterowanie trajektorią efektora w przestrzeni zadaniowej (np. X_0, Y_0, Z_0), przekształcając sterowanie na poziom przestrzeni przegubów q . Poniżej przedstawiono szczegółowe równania wykorzystując podejście przedstawione w artykule [2] opisujące proces od wejścia do wyjścia. Relacja kinematyczna opisuje położenie efektora w przestrzeni zadaniowej jako funkcję konfiguracji manipulatora

$$y_i = k_i(q), \quad i = 1, 2, \dots, n, \quad (2.3)$$

gdzie y_i to współrzędne wyjścia, a q to wektor położenia przegubów. Różniczkując równanie kinematyczne względem czasu, uzyskujemy wyrażenie na prędkości efektora

$$\dot{y}_i = J_i(q)\dot{q}, \quad (2.4)$$

gdzie $J_i(q)$ to macierz Jacobiego, opisująca zależność między prędkościami przegubów a prędkościami efektora. Dalsze różniczkowanie względem czasu prowadzi do wyrażenia

$$\ddot{y}_i = \frac{d^2}{dt^2}k_i(q) = \dot{J}_i(q)\dot{q} + J_i(q)\ddot{q} = \dot{q}^T \frac{\partial^2 k_i}{\partial q^2} \dot{q} + J_i(q)\ddot{q} = P_i + J_i\ddot{q}. \quad (2.5)$$

jest nieliniowym wyrażeniem związanym z pochodnymi drugiego rzędu.

Łącząc wszystkie współrzędne wyjściowe w jeden wektor, uzyskujemy

$$\ddot{y} = P + J(q)\ddot{q}, \quad (2.6)$$

gdzie \ddot{y} to wektor przyspieszeń efektora, P to nieliniowe wyrażenie, a $J(q)$ to macierz Jacobiego.

Dynamiczny model manipulatora w przestrzeni przegubów opisuje zależności między siłami wejściowymi a przyspieszeniami przegubów

$$M(q)\ddot{q} + C(q, \dot{q})\dot{q} + D(q) + T(\dot{q}) = u, \quad (2.7)$$

gdzie:

- $M(q)$ – macierz mas (symetryczna i dodatnio określona),
- $C(q, \dot{q})$ – macierz sił Coriolisa i odśrodkowych,
- $D(q)$ – wektor sił grawitacyjnych,
- $T(\dot{q})$ – siły tarcia w przegubach,
- u – wektor wejść sterujących,
- $q = [q_1, q_2, q_3]^T$ – wektor kątów przegubów,

Podstawiając (2.7) do (2.6), otrzymujemy

$$\ddot{y} = P + J(q)M^{-1}(u - C(q, \dot{q})\dot{q} - D(q) - T(\dot{q})), \quad (2.8)$$

co rozwijając, daje

$$\ddot{y} = P - JM^{-1}C\dot{q} - JM^{-1}D - JM^{-1}T + JM^{-1}u. \quad (2.9)$$

Równanie (2.9) opisuje afiniczny układ sterowania z następującym równaniem

$$\ddot{y} = F + Gu, \quad (2.10)$$

gdzie

$$F = P - JM^{-1}C\dot{q} - JM^{-1}D - JM^{-1}T, \quad (2.11)$$

$$G = JM^{-1}. \quad (2.12)$$

Zakładając, że macierz G jest odwracalna. Wprowadzamy (2.13) do układu afinicznego

$$u = G^{-1}(-F + v), \quad (2.13)$$

gdzie v to nowe wejście do układu. Podstawiając sterowanie (2.13) do układu (2.10) otrzymujemy układ zamkniętej pętli sprzężenia zwrotnego wyrażony w postaci postaci podwójnego liniowego integratora z nowym wejściem

$$\ddot{y} = v. \quad (2.14)$$

Aby zrealizować żadaną trajektorię $y_d(t)$, stosuje się regulator PD z korekcją w przestrzeni zadaniowej

$$v = \ddot{y}_d - K_d(\dot{y} - \dot{y}_d) - K_p(y - y_d), \quad (2.15)$$

gdzie:

- $y_d, \dot{y}_d, \ddot{y}_d$ – żądane położenie, prędkość i przyspieszenie,
- K_p, K_d – macierze wzmocnień proporcjonalnych i różniczkujących.

Dzięki zastosowaniu odsprzęgania wejścia-wyjścia możliwe jest niezależne sterowanie każdym wyjściem układu, eliminując sprzężenia między przegubami manipulatora.

Rozdział 3

Sterowanie predykcyjne

W tym rozdziale omówiono sterowanie predykcyjne oparte na modelu (*Model Predictive Control*, MPC) w kontekście identyfikacji parametrów manipulatora 3R, takich jak masy segmentów oraz długości ramion. Przedstawiono podstawy teoretyczne metody, sposób jej zastosowania do identyfikacji parametrów oraz opisano implementację algorytmu MPC w tym celu. Do opisanego metody sterowania predykcyjnego opartego na modelu wykorzystano artykuł [1].

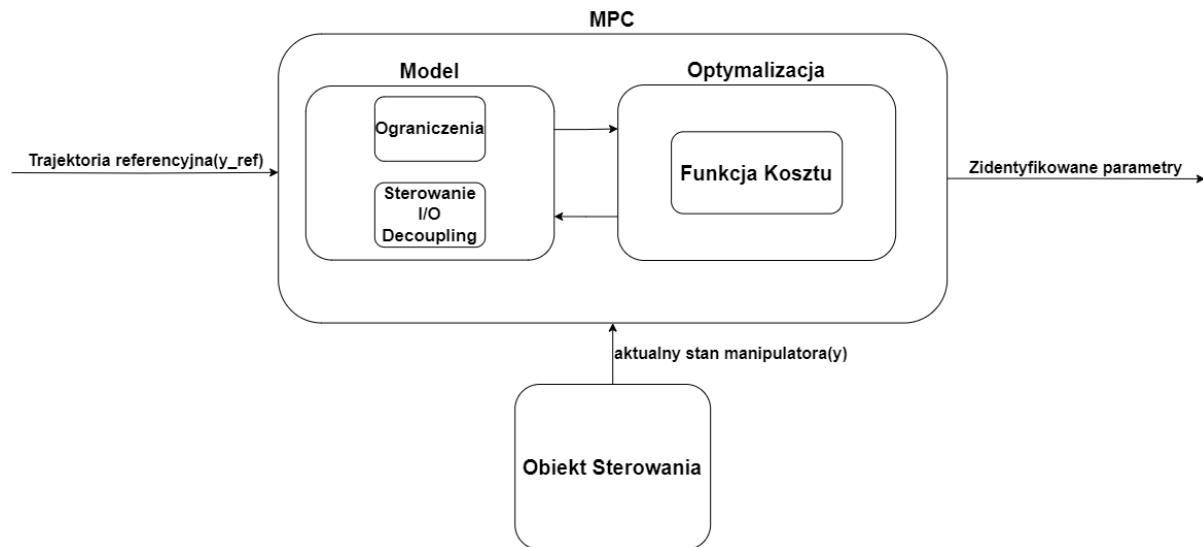
3.1 Opis metody MPC

Sterowanie predykcyjne oparte na modelu jest zaawansowaną metodą sterowania i optymalizacji, która wykorzystuje model obiektu do przewidywania jego przyszłego zachowania. W kontekście identyfikacji parametrów, MPC może być użyte do estymacji nieznanych parametrów modelu poprzez minimalizację różnicy między zachowaniem przewidywanym przez model a rzeczywistymi obserwacjami. Sterowanie predykcyjne oparte na modelu (MPC) jest szeroko omawiane w literaturze, w tym w książkach [4] i [6].

Główne kroki w metodzie MPC są następujące:

1. **Predykcja:** Wykorzystując aktualny stan układu oraz model dynamiki, przewiduje się przyszłe stany układu w horyzoncie predykcji.
2. **Optymalizacja:** Rozwiązuje się problem optymalizacji, minimalizując funkcję kosztu, która uwzględnia odchylenie od trajektorii referencyjnej oraz nakłady na sterowanie.
3. **Implementacja sterowania:** Z optymalnej sekwencji sterowań zastosowane jest tylko pierwsze sterowanie, po czym cały proces jest powtarzany w następnym kroku czasowym.

Schemat blokowy procesu identyfikacji parametrów z wykorzystaniem MPC przedstawiono na rysunku 3.1. Blok MPC zawiera model manipulatora z ustalonymi parametrami oraz algorytm optymalizacji służący do estymacji parametrów na podstawie różnicy między trajektorią przewidywaną a referencyjną.



Rysunek 3.1 Blokowy schemat identyfikacji parametrów z wykorzystaniem MPC

3.2 Wymagania MPC wobec optymalizatora

Sterowanie MPC jest w istocie problemem optymalizacji, gdzie w każdym kroku czasowym minimalizowana jest funkcja kosztu przy uwzględnieniu ograniczeń. Wymaga to zastosowania wydajnych metod optymalizacji numerycznej, zdolnych do szybkiego rozwiązywania problemów w czasie rzeczywistym. W kontekście zastosowań MPC z ograniczeniami, istotne informacje można znaleźć w książce [3].

MPC wymaga od optymalizatora:

- **Szybkości działania:** Optymalizator musi znajdować rozwiązania w krótkim czasie, aby sterowanie było aktualne i efektywne.
- **Radzenia sobie z nieliniowością:** W przypadku nieliniowych modeli, optymalizator musi być zdolny do rozwiązywania nieliniowych problemów optymalizacyjnych.
- **Obsługi ograniczeń:** Optymalizator musi uwzględniać ograniczenia na stany i sterowania, takie jak ograniczenia fizyczne manipulatora czy bezpieczeństwa.
- **Stabilności i niezawodności:** Powinien gwarantować zbieżność do optymalnego rozwiązania i być odporny na błędy numeryczne.

Do rozwiązywania problemu optymalizacji w MPC stosuje się różne metody:

- **Programowanie kwadratowe (QP)** – dla modeli liniowych i funkcji kosztu kwadratowej.
- **Metody gradientowe** – dla modeli nieliniowych.
- **Algorytmy ewolucyjne** – dla problemów złożonych lub z wieloma ekstremami lokalnymi.

W implementacji sterowania manipulatorem 3R zastosowano metodę optymalizacji ewolucyjnej, taką jak algorytm różnicowej ewolucji, ze względu na nieliniowość modelu i konieczność optymalizacji parametrów manipulatora.

3.3 Formułowanie problemu identyfikacji parametrów

Identyfikacja parametrów dynamicznych manipulatora polega na estymacji kluczowych wielkości fizycznych, takich jak długości segmentów, masy czy współczynniki tarcia, które determinują jego dynamikę. Proces ten realizowany jest za pomocą algorytmów sterowania predykcyjnego opartego na modelu (z ang. *Model Predictive Control, MPC*), minimalizujących funkcję kosztu wyrażającą różnicę między zachowaniem przewidywanym przez model a rzeczywistymi pomiarami.

Problem identyfikacji parametrów można sformułować jako zadanie optymalizacji, polegające na minimalizacji funkcji kosztu J , opisanej równaniem (3.1)

$$J(u) = \sum_{k=0}^{N_{pred}-1} \left(f(x_k, x_k^{ref}) + f(u_k, u_k^{ref}) \right) \quad (3.1)$$

gdzie:

- $J(u)$: funkcja kosztu sumująca błędy trajektorii i zużycie energii,
- θ – wektor nieznanych parametrów modelu (np. masy, długości segmentów),
- x_k : przewidywany wektor pozycji w kroku k .
- x_k^{ref} : Stan referencyjny (pomiar rzeczywisty) w kroku k .
- u_k^{ref} : Referencyjny wektor sterowań w kroku k .
- u_k : Przewidywany wektor sterowania w kroku k .
- $f(x_k, x_k^{ref})$: Oblicza różnicę między przewidywaną pozycją x_k a pozycją x_k^{ref} . Może to być np. odległość euklidesowa lub odchylenie kątowe.
- $f(u_k, u_k^{ref})$: Wyznacza różnicę między przewidywanym sterowaniem u_k a sterowaniem referencyjnym u_k^{ref} .
- N_{pred} : Liczba kroków w przyszłość uwzględniana w optymalizacji.
- $\theta_{min}, \theta_{max}$ – dolne i górne ograniczenia na wartości parametrów.

Celem identyfikacji jest znalezienie takich wartości wektora parametrów θ , które minimalizują funkcję kosztu J . Model dynamiki manipulatora $f(x_k, u_k, \theta)$ umożliwia przewidywanie jego zachowania na podstawie aktualnego stanu oraz zadanych parametrów i sterowań. Funkcja kosztu uwzględnia błędy trajektorii oraz ograniczenia na sterowanie, co pozwala na precyzyjne identyfikowanie parametrów rzeczywistych manipulatora.

Algorytm MPC iteracyjnie rozwiązuje problem optymalizacji poprzez następujące kroki:

1. Przewidywanie trajektorii $x_k(\theta)$ na podstawie aktualnego stanu x_0 oraz estymowanych parametrów θ .
2. Obliczenie wartości funkcji kosztu J na podstawie różnicy między trajektorią przewidywaną a referencyjną x_k^{ref} .
3. Aktualizację parametrów θ w kierunku minimalizacji funkcji kosztu.

Ostatecznym celem procesu jest identyfikacja takich wartości parametrów θ , które minimalizują błędy trajektorii, zapewniając jednocześnie zgodność modelu z rzeczywistym zachowaniem manipulatora.

3.4 Horyzont predykcji

Horyzont predykcji jest kluczowym parametrem w metodzie sterowania predykcyjnego opartego na modelu. Horyzont predykcji T określa czas, na jaki system przewiduje przyszłe stany w procesie optymalizacji sterowania predykcyjnego. Jest on definiowany jako iloczyn liczby kroków predykcji N_{pred} oraz czasu próbkowania dt

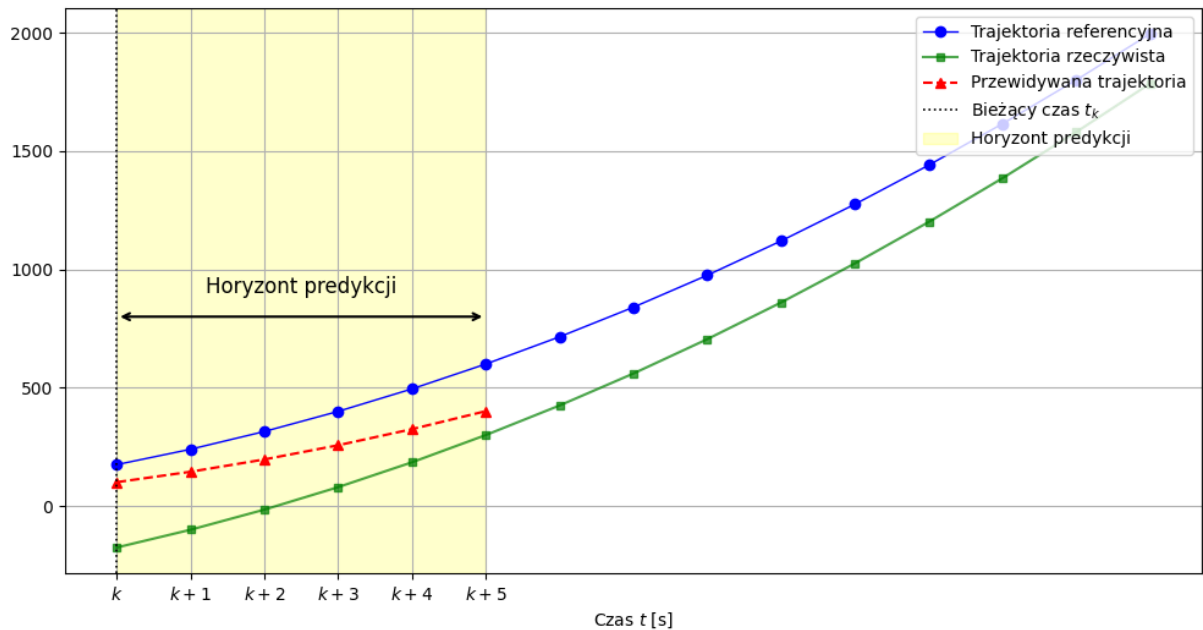
$$T = N_{\text{pred}} \cdot dt. \quad (3.2)$$

W kontekście identyfikacji parametrów manipulatora, odpowiedni dobór T pozwala lepiej uwzględnić długoterminowe konsekwencje działania sterowania oraz redukować błędy położenia i prędkości.

Długość horyzontu predykcji wpływa na:

- **Dokładność sterowania:** Zbyt krótki T może prowadzić do sytuacji, w której optymalizacja ignoruje odległe konsekwencje sterowania, co skutkuje mniej precyzyjnym odwzorowaniem trajektorii referencyjnej.
- **Złożoność obliczeniową:** Zwiększenie T wymaga większych nakładów obliczeniowych, ponieważ liczba zmiennych w optymalizacji rośnie proporcjonalnie do długości horyzontu.

Przykład schematu horyzontu predykcji przedstawiono na rysunku 3.2, gdzie bieżący stan układu x_k jest używany do przewidywania przyszłych stanów $x_{k+1}, x_{k+2}, \dots, x_{k+N_{\text{pred}}}$. Proces ten uwzględnia zarówno model dynamiki manipulatora, jak i ograniczenia nałożone na stan oraz sterowanie.



Rysunek 3.2 Schemat horyzontu predykcji w MPC

W praktyce dobór długości horyzontu predykcji jest kompromisem między jakością sterowania a wymaganiami obliczeniowymi. Na przykład dla systemów dynamicznych o wolnej dynamice (np. procesy chemiczne) zaleca się stosowanie dłuższych horyzontów

predykcji, w przypadku manipulatorów robotycznych, takich jak manipulator 3R, długość horyzontu predykcji nie może być zbyt duża, aby pozwolić na precyzyjne sterowanie przy akceptowalnym czasie obliczeń.

Warto również zauważyć, że czas obliczeń dla dłuższych horyzontów predykcji może być ograniczeniem w systemach działających w czasie rzeczywistym. Dlatego należy dokładnie analizować, jak długość horyzontu predykcji wpływa na czas obliczeń oraz jakość identyfikacji parametrów.

3.5 Dobór współczynników funkcji kosztu i ograniczeń parametrów

W procesie implementacji sterowania predykcyjnego opartego na modelu kluczowym aspektem jest odpowiedni dobór współczynników funkcji kosztu oraz określenie realistycznych ograniczeń dla parametrów modelu. Te decyzje mają istotny wpływ na jakość sterowania, stabilność algorytmu oraz skuteczność identyfikacji parametrów.

Funkcja kosztu w algorytmie MPC jest definiowana jako suma dwóch składników: błędu trajektorii oraz nakładów na sterowanie. Każdy z tych składników jest ważony współczynnikami, które określają priorytet tych dwóch celów.

Dobór współczynników jest uzależniony od specyfiki aplikacji i wymaga kompromisu między dwoma celami. Współczynnik przy $f(x_k, x_k^{ref})$ odpowiada za minimalizację błędu trajektorii, co jest szczególnie istotne w zadaniach wymagających precyzyjnego śledzenia trajektorii, takich jak manipulacja w precyzyjnych systemach. Z kolei współczynnik przy $f(u_k, u_k^{ref})$ koncentruje się na minimalizacji nakładów na sterowanie, co jest korzystne w systemach, gdzie kluczowe znaczenie mają oszczędność energetyczna lub ograniczenie obciążeń mechanicznych. Ostatecznie, najczęściej dobiera się wartości obu współczynników w taki sposób, aby osiągnąć równowagę pomiędzy dokładnością sterowania a efektywnością energetyczną.

Optymalny dobór współczynników może być uzyskiwany na podstawie analizy wrażliwości funkcji kosztu na ich wartości, co pozwala zrozumieć wpływ poszczególnych wag na wynikowe sterowanie. Sterowanie predykcyjne opiera się na optymalizacji trajektorii z uwzględnieniem ograniczeń, które mogą być twarde (nienaruszalne) lub miękkie (mogą być przekroczone kosztem dodatkowego ograniczenia).

- **Ograniczenia twarde:** Gwarantują bezpieczeństwo działania manipulatora i spełnienie krytycznych wymagań, takich jak maksymalne przyspieszenie, prędkość czy zakres ruchu przegubów. Niemniej jednak ich zastosowanie może prowadzić do problemów z zbieżnością algorytmu, jeśli optymalizacja stanie się nieosiągalna.
- **Ograniczenia miękkie:** Umożliwiają większą elastyczność w sterowaniu poprzez wprowadzenie kar w funkcji kosztu za przekroczenie ograniczeń. Takie podejście zwiększa skuteczność identyfikacji parametrów, szczególnie w przypadku modeli z wysoką nieliniowością.

Podczas identyfikacji parametrów, takich jak długości segmentów czy masy ogniw manipulatora, istotne jest określenie realistycznych ograniczeń dla tych wielkości. Ograniczenia te zapobiegają znalezieniu nieprawidłowych rozwiązań przez algorytm optymalizacji i zwiększają jego stabilność. Zakresy parametrów, takich jak długości segmentów czy masy, można określić na podstawie konstrukcji fizycznej, analizy literaturowej oraz eksperymentalnych pomiarów.

Przedziały parametrów powinny odzwierciedlać realistyczne wartości wynikające z projektu manipulatora, a dane techniczne lub wcześniejsze badania mogą dostarczyć przybliżonych przedziałów dla podobnych układów. W przypadku dostępności rzeczywistego manipulatora możliwe jest także wyznaczenie dokładnych wartości parametrów poprzez bezpośredni pomiar.

Brak realistycznych ograniczeń parametrów może prowadzić do nierealistycznych wyników optymalizacji, takich jak fizycznie niemożliwe wartości, lub wydłużenia czasu obliczeń, gdyż szersze przedziały zwiększają złożoność przestrzeni poszukiwań. Określenie odpowiednich ograniczeń pozwala na szybsze i bardziej precyzyjne uzyskanie wyników optymalizacji.

Rozdział 4

Implementacja algorytmu MPC

W tym rozdziale przedstawiono implementację algorytmu sterowania predykcyjnego opartego na modelu (MPC) użytego do identyfikacji parametrów manipulatora 3R. Algorytm został zaimplementowany w języku Python, co umożliwia elastyczne przeprowadzanie symulacji oraz łatwe wprowadzanie modyfikacji. Opisano strukturę kodu, funkcję kosztu, procedurę predykcji trajektorii oraz zastosowaną metodę optymalizacji parametrów. W odróżnieniu od klasycznego zastosowania MPC do sterowania, w tej implementacji algorytm został użyty wyłącznie do identyfikacji parametrów modelu, bez bezpośredniego wpływu na sterowanie manipulatorem.

Implementacja algorytmu MPC została podzielona na kilka modułów, z których każdy pełni określoną funkcję:

- `modelODE.py` – zawiera funkcje odpowiedzialne za obliczenia związane z dynamiką manipulatora, takie jak macierz mas, macierz Coriolisa oraz wektor sił grawitacyjnych. Model ten jest kluczowy dla dokładnej predykcji zachowania manipulatora i umożliwia estymację parametrów na podstawie obserwowanych danych.
- `effectorTrajectoryGenerator3D.py` – moduł generujący trajektorie referencyjne efektora, takie jak trajektoria eliptyczna czy trajektoria w kształcie księżyca. Te trajektorie służą jako odniesienie podczas procesu identyfikacji, umożliwiając porównanie przewidywanego zachowania z rzeczywistym.
- `mpc.py` – główny moduł implementujący algorytm MPC, w tym funkcję kosztu, procedurę predykcji trajektorii oraz optymalizację parametrów. W tej implementacji MPC jest wykorzystywane do estymacji parametrów modelu manipulatora na podstawie minimalizacji różnicy między przewidywaną a referencyjną trajektorią.
- `start.py` – skrypt startowy uruchamiający symulację i zapis wyników. Określa parametry początkowe symulacji, inicjuje pętlę czasową dla algorytmu MPC oraz obsługuje zbieranie danych niezbędnych do procesu identyfikacji.

4.1 Funkcja kosztu

Funkcja kosztu uwzględnia zarówno błędy pozycji między przewidywaną a referencyjną trajektorią, jak i zużycie energii na sterowanie. Dla każdego kroku czasowego obliczany jest błąd pozycji oraz błąd związany z nakładami na sterowanie. Wagi dla tych błędów regulowane są przez parametry λ_u oraz λ_e .

$$\min_{\theta} J = \frac{1}{N_{\text{pred}}} \left(\lambda_e \sum_{k=1}^{N_{\text{pred}}} \|x_k(\theta) - x_k^{\text{ref}}\|_2 + \lambda_u \sum_{k=1}^{N_{\text{pred}}} \|\dot{x}_k(\theta) - \dot{x}_k^{\text{ref}}\|_2 \right), \quad (4.1)$$

przy warunkach:

$$x_{k+1} = f(x_k, u_k, \theta),$$

$$\theta_{\min} \leq \theta \leq \theta_{\max},$$

gdzie:

- θ – wektor nieznanych parametrów modelu (np. masy, długości segmentów),
- $x_k(\theta)$ – przewidywany wektor pozycji w kroku k dla danych parametrów θ ,
- $\dot{x}_k(\theta)$ – przewidywany wektor prędkości w kroku k dla danych parametrów θ ,
- $x_k^{\text{ref}}, \dot{x}_k^{\text{ref}}$ – odpowiednio pozycje i prędkości referencyjne, które układ powinien śledzić,
- λ_e, λ_u – Współczynniki błędu pozycji i prędkości,
- N_{pred} – horyzont predykcji,
- $\theta_{\min}, \theta_{\max}$ – dolne i górne ograniczenia na wartości parametrów.

4.2 Predykcja trajektorii

Predykcja trajektorii jest kluczowym elementem algorytmu MPC. Funkcja `calculate_trajectory` oblicza przewidywaną trajektorię na podstawie aktualnych parametrów oraz bieżącego stanu manipulatora, używając solvera ODE do numerycznego rozwiązywania równań ruchu. Trajektorja jest przewidywana na horyzoncie czasowym N_{pred} , co pozwala na analizę przyszłych stanów manipulatora. Model dynamiki manipulatora wykorzystywany w tej funkcji uwzględnia macierz mas, siły Coriolisa oraz wpływ grawitacji.

Pseudokod funkcji predykcji trajektorii został przedstawiony w algorytmie (1)

```
function OBLICZ_TRAJEKTORIE(biezący_stan, parametry,  $N_{\text{pred}}$ ,  $dt$ , czas_startowy)
    Utwórz tablicę na przewidywaną trajektorię
    Ustaw solver do rozwiązywania równań dynamicznych
    while solver jest skuteczny and czas < czas_startowy +  $N_{\text{pred}} \cdot dt$  do
        Oblicz stan dla kolejnego kroku czasowego
        Dodaj nowy stan do przewidywanej trajektorii
    end while
    return przewidywana trajektorja
end function
```

Algorytm 1: Funkcja predykcji trajektorii

4.3 Optymalizacja parametrów

Głównym celem algorytmu MPC w tej implementacji jest identyfikacja parametrów modelu, takich jak masy i długości segmentów manipulatora. W tym celu zastosowano metodę różnicowej ewolucji (z ang. *differential evolution*), która sprawdza się w problemach nieliniowych. Procedura optymalizacji minimalizuje funkcję kosztu poprzez iteracyjne dostosowanie parametrów modelu.

Pseudokod procesu optymalizacji został przedstawiony w algorytmie (2)

```
function OPTYMALIZUJ_PARAMETRY(parametry początkowe, trajektoria referencyj-  
na)  
    Zainicjalizuj parametry optymalizacji i ograniczenia  
    while nie osiągnięto zbieżności do  
        Przewidź trajektorię za pomocą oblicz_trajektorię  
        Oblicz funkcję kosztu  
        Zaktualizuj parametry za pomocą ewolucji różnicowej  
    end while  
    return zoptymalizowane parametry  
end function
```

Algorytm 2: Funkcja optymalizacji parametrów

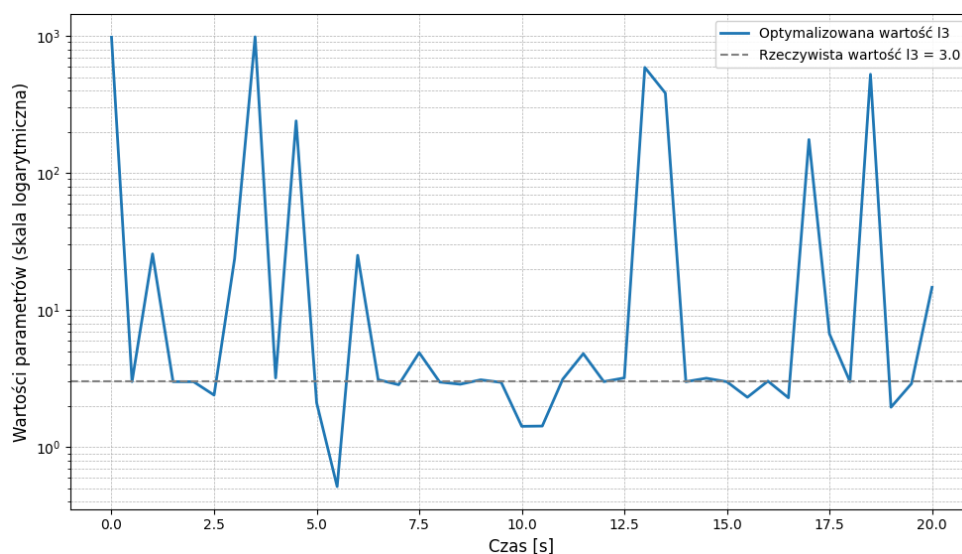
Rozdział 5

Wyniki symulacji i analiza wyników

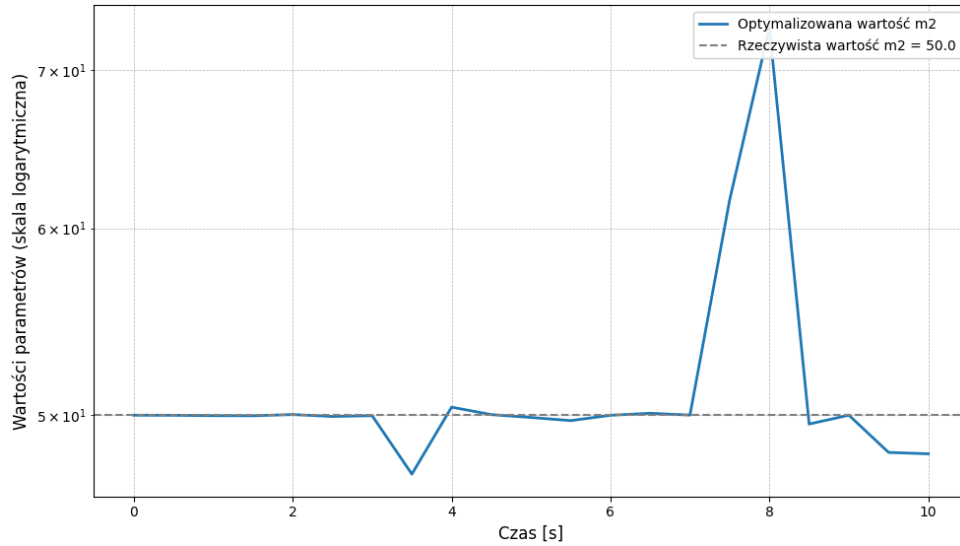
W niniejszym rozdziale przedstawiono wyniki symulacji przeprowadzonych w celu identyfikacji parametrów manipulatora z wykorzystaniem sterowania predykcyjnego opartego na modelu. Analizowano różne wartości parametrów symulacji, w tym wagi funkcji kosztu (λ_U, λ_E), ilość kroków predykcji (N_{pred}), czas próbkowania dt oraz różne zestawy parametrów identyfikowanych ($m_1, m_2, m_3, l_1, l_2, l_3$). Szczególną uwagę zwrócono na różnice wyników uzyskanych przy uwzględnieniu ograniczeń oraz bez ich uwzględnienia. Wyniki przedstawiono w tabelach oraz na wykresach, które ilustrują zbieżność optymalizowanych parametrów z rzeczywistymi wartościami.

5.1 Wpływ ograniczeń na dokładność identyfikacji parametrów

Jak pokazano na rysunku 5.1, brak ograniczeń długości ogniwa powoduje znaczną rozbieżność estymowanych parametrów w stosunku do rzeczywistych wartości. Wprowadzenie ograniczeń (rysunek 5.2) pozwala na uzyskanie wyników bardziej zgodnych z oczekiwaniami, co podkreśla znaczenie realistycznych przedziałów parametrów w optymalizacji.



Rysunek 5.1 Brak ograniczeń długości ogniwa.



Rysunek 5.2 Zaimplementowane ograniczenia.

Zastosowanie ograniczeń na wartości parametrów okazało się kluczowe dla poprawy dokładności wyników, eliminując rozbieżności estymowanych parametrów względem ich rzeczywistych wartości.

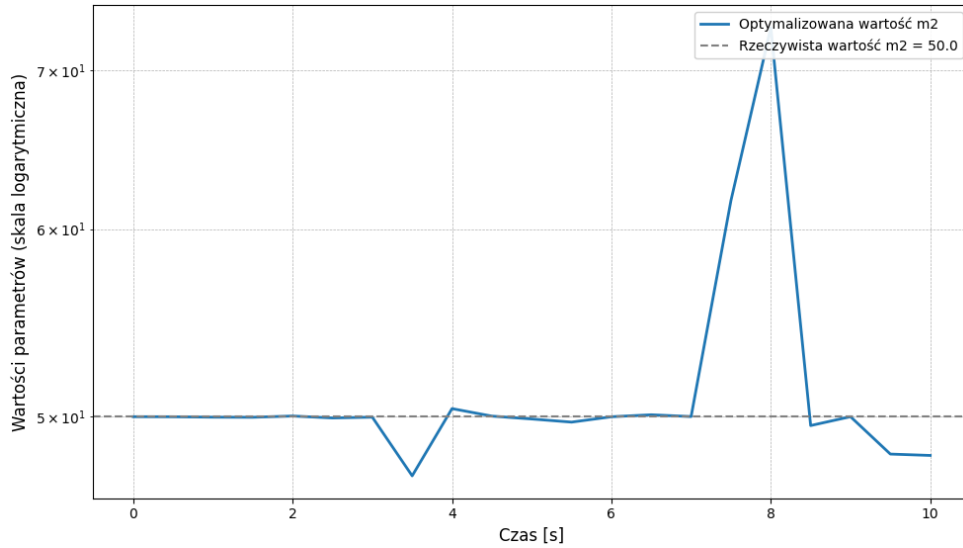
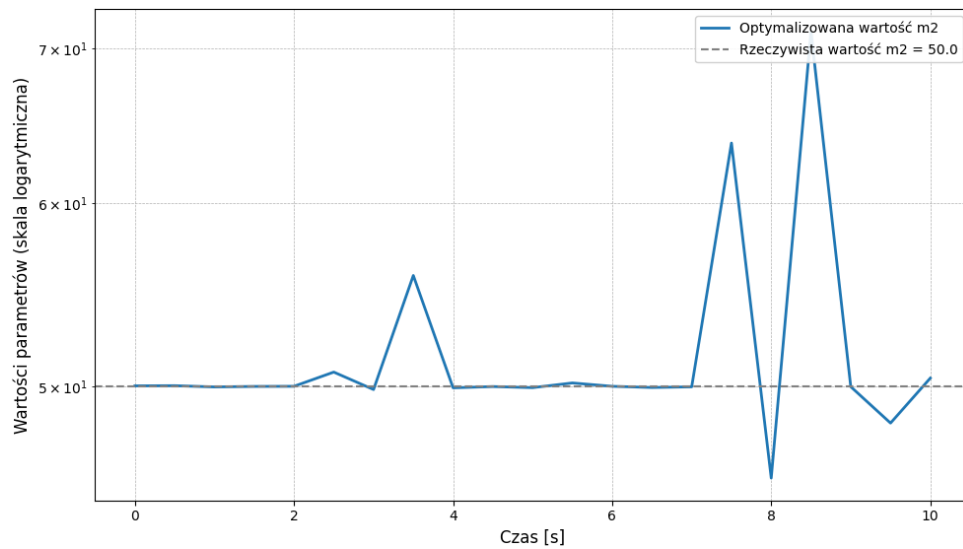
5.2 Dobór współczynników funkcji kosztu

Wykonano szereg symulacji z różnymi wartościami współczynników funkcji kosztu (λ_E, λ_U), aby zbadać ich wpływ na dokładność identyfikacji parametrów manipulatora. Tabela 5.1 przedstawia wyniki dla parametru m_2 , a ilustracje na rysunkach 5.3 i 5.4 przedstawiają przebiegi uzyskane w wybranych wartościach współczynników.

Na podstawie wyników można zauważyć, że najlepsze rezultaty w identyfikacji parametrów manipulatora uzyskano, gdy funkcja kosztu uwzględniała jedynie błąd trajektorii ($\lambda_E = 1, \lambda_U = 0$). Dodanie błędów nakładów sterowań do funkcji kosztów ($\lambda_U > 0$) wprowadzało dodatkowe błędy, co wskazuje na mniejszą skuteczność tego podejścia w analizowanym przypadku.

λ_E	λ_U	m_2	Błąd
1.0	0.0	50.0001	0.0001
0.9	0.1	49.9984	0.0016
0.8	0.2	50.0004	0.0004
0.7	0.3	50.0004	0.0004
0.6	0.4	49.9867	0.0132
0.5	0.5	50.0266	0.0266
0.4	0.6	49.9878	0.0121
0.3	0.7	49.9878	0.0121
0.2	0.8	49.9878	0.0121
0.1	0.9	49.9867	0.0132
0.0	1.0	49.9955	0.0044

Tabela 5.1 Wpływ współczynników funkcji kosztu na dokładność identyfikacji parametrów

Rysunek 5.3 Współczynnik λ_E równe 1 oraz λ_U równe zeroRysunek 5.4 Współczynnik λ_E równe 1 oraz λ_U równe 1

Optymalne rezultaty osiągnięto przy uwzględnieniu jedynie błędu trajektorii w funkcji kosztu, co podkreśla znaczenie trafnego wyboru parametrów optymalizacyjnych. Dodanie terminu uwzględniającego nakłady sterowania negatywnie wpłynęło na dokładność, wskazując, że w analizowanym przypadku bardziej efektywne było skupienie się wyłącznie na precyzji trajektorii.

5.3 Wpływ ilości kroków predykcji

Tabela 5.2 przedstawia wyniki symulacji przeprowadzonych dla różnych wartości liczby kroków predykcji (N_{pred}). Analiza wyników pokazuje, że liczba kroków predykcji ma istotny wpływ na dokładność identyfikacji parametru m_2 .

Przy $N_{\text{pred}} = 10$ uzyskano najlepsze wyniki, co wskazuje na optymalną wartość dla tego parametru. Zwiększenie liczby kroków predykcji powyżej tej wartości prowadzi do wyraźnego spadku dokładności identyfikacji parametrów. Jest to spowodowane większą

złożonością problemu optymalizacji, co może utrudniać algorytmowi znalezienie dokładnych rozwiązań.

N_{pred}	m_2	Błąd
1	49.9999	0.0001
5	49.9956	0.0043
10	50.0001	0.0001
15	50.0088	0.0088
20	49.5776	0.4223
25	47.0176	2.9823

Tabela 5.2 Wpływ ilości kroków predykcji na dokładność identyfikacji parametrów

Najlepsze wyniki identyfikacji uzyskano dla liczby kroków predykcji mniejszej lub równej dziesięciu, co wskazuje na optymalną długość horyzontu predykcji. Zwiększanie liczby kroków predykcji prowadziło do obniżenia dokładności z powodu złożoności problemu optymalizacji.

5.4 Wpływ czasu próbkowania

Tabela 5.3 prezentuje wpływ czasu próbkowania (dt) na dokładność identyfikacji parametru m_2 . Na podstawie wyników można zauważyć, że wybór odpowiedniego czasu próbkowania jest kluczowy dla uzyskania precyzyjnych wyników identyfikacji.

Dla małych wartości dt (np. $dt = 0.1$) uzyskano wysoką dokładność identyfikacji, ale kosztem zwiększenia złożoności obliczeniowej. Optymalnym kompromisem okazał się czas próbkowania $dt = 0.5$, który zapewnia minimalny błąd przy akceptowalnym czasie obliczeń. Natomiast zwiększanie dt powyżej 0.5 prowadzi do gwałtownego wzrostu błędu, co wskazuje, że większy czas próbkowania negatywnie wpływa na dokładność identyfikacji.

dt	m_2	Błąd
0.1	49.9999	0.0001
0.25	50.0009	0.0009
0.5	50.0001	0.0001
0.75	49.9797	0.0202
1.0	49.9309	0.0690
2.0	43.1281	6.8718
10.0	46.3605	3.6394

Tabela 5.3 Wpływ czasu próbkowania na dokładność identyfikacji parametrów

5.5 Wpływ dokładności optymalizatora

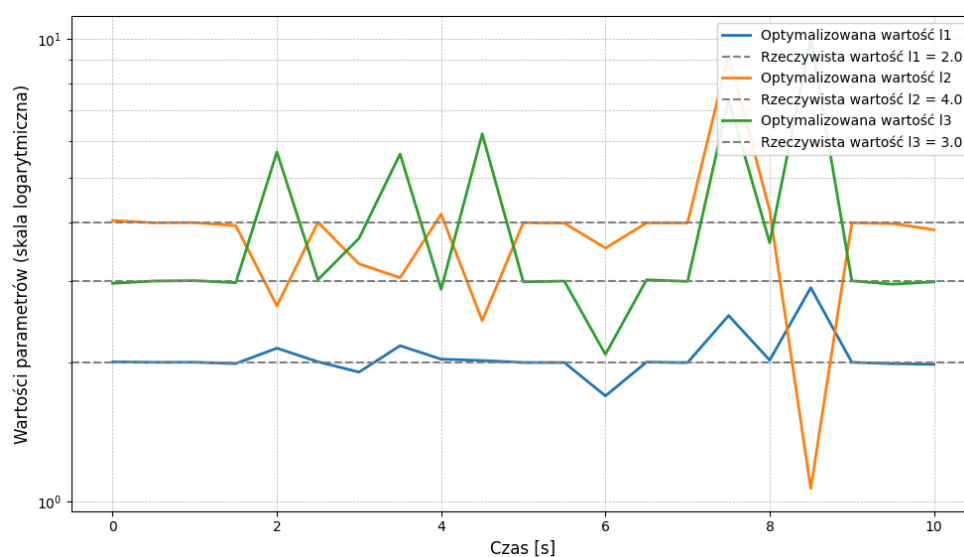
Tabela 5.4 przedstawia wpływ liczby iteracji optymalizatora ($maxiter$) oraz tolerancji (tol) na wyniki identyfikacji parametru m_2 . Z analizy wynika, że zwiększanie $maxiter$ i zmniejszanie tol poprawia dokładność identyfikacji. Jednak nadmierne zwiększanie liczby iteracji może prowadzić do nieefektywnego wykorzystania zasobów obliczeniowych. Najlepsze wyniki uzyskano dla $maxiter = 100$ i $tol = 1e^{-9}$, co stanowi kompromis między dokładnością a czasem obliczeń.

$maxiter$	tol	m_2	Błąd
1	$1e^{-3}$	49.9702	0.0297
10	$1e^{-6}$	50.0245	0.0245
100	$1e^{-9}$	50.0001	0.0001
1000	$1e^{-12}$	50.0013	0.0013
10000	$1e^{-15}$	50.0009	0.0009

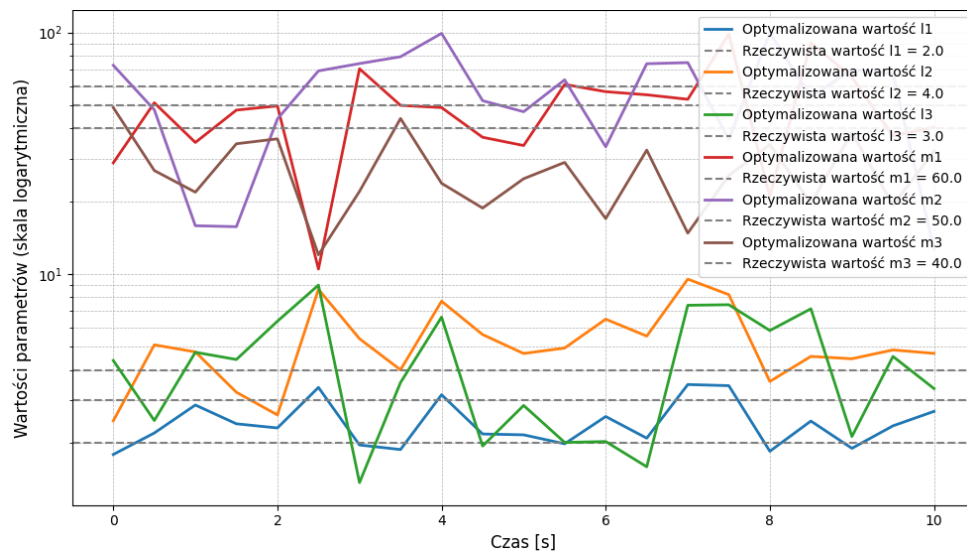
Tabela 5.4 Wpływ ilości iteracji oraz tolerancji optymalizacji na dokładność identyfikacji parametrów

5.6 Wpływ ilości badanych parametrów na dokładność

Symulacje przeprowadzone dla różnych konfiguracji liczby parametrów uwzględnianych w procesie identyfikacji wykazały, że wraz ze wzrostem liczby identyfikowanych parametrów ($m_1, m_2, m_3, l_1, l_2, l_3$) złożoność problemu optymalizacji znacząco wzrasta. Wiąże się to z wydłużeniem czasu obliczeń oraz zmniejszeniem skuteczności algorytmu. Dodatkowo, większa liczba parametrów powoduje wzrost wrażliwości funkcji kosztu na niedokładności modelu, co może skutkować większymi błędami w identyfikacji poszczególnych parametrów. Na rysunku 5.5 przedstawiono zbieżność identyfikowanych parametrów do rzeczywistych wartości dla trzech parametrów, a na rysunku 5.6 zilustrowano zbieżność identyfikowanych parametrów do wartości rzeczywistych dla sześciu parametrów.



Rysunek 5.5 Symulacja dla 3 parametrów (l_1, l_2, l_3)



Rysunek 5.6 symulacja dla 6 parametrów ($m_1, m_2, m_3, l_1, l_2, l_3$)

Rozdział 6

Podsumowanie

Przeprowadzone badania potwierdziły, że metoda sterowania predykcyjnego opartego na modelu jest skutecznym narzędziem do identyfikacji parametrów manipulatora. Odpowiedni dobór parametrów algorytmu, takich jak horyzont predykcji, czas próbkowania oraz wagi funkcji kosztu, pozwala na uzyskanie wysokiej dokładności identyfikacji przy jednoczesnym zachowaniu efektywności obliczeniowej. Szczególne znaczenie miało zastosowanie ograniczeń na wartości parametrów, co wyeliminowało błędy wynikające z fizycznie nierealistycznych estymacji i znacząco poprawiło zbieżność wyników z rzeczywistymi wartościami. Wyniki symulacji wykazały, że optymalnym kompromisem była konfiguracja, w której czas próbkowania wynosił $dt = 0.5$, liczba kroków predykcji była równa $N_{\text{pred}} = 10$, a funkcja kosztu skupiała się głównie na błędzie trajektorii, co oznaczało przypisanie wartości $\lambda_E = 1$ i $\lambda_U = 0$. Analiza wykazała także, że wzrost liczby jednocześnie identyfikowanych parametrów wiązał się ze zwiększoną złożonością obliczeniową oraz większym ryzykiem błędów, co podkreśla konieczność ograniczania liczby estymowanych parametrów w celu zapewnienia stabilności procesu.

Osiągnięte wyniki otwierają nowe kierunki dalszych badań, które mogą przyczynić się do rozwoju metod identyfikacji parametrów manipulatorów oraz rozszerzenia zastosowań sterowania predykcyjnego. Pierwszym istotnym obszarem dalszych badań jest integracja metod sztucznej inteligencji, w szczególności algorytmów uczenia maszynowego, takich jak uczenie ze wzmacnianiem, które mogą dynamicznie optymalizować parametry algorytmu MPC w czasie rzeczywistym. Drugim kierunkiem jest rozszerzenie modelu manipulatora poprzez uwzględnienie dodatkowych czynników, takich jak tarcie w przegubach, elastyczność elementów czy wpływ zakłóceń zewnętrznych, co pozwoli na lepsze odwzorowanie rzeczywistych warunków pracy. Kolejnym krokiem powinno być przeprowadzenie badań eksperymentalnych na rzeczywistych manipulatorach w celu weryfikacji skuteczności metody w praktyce oraz adaptacji algorytmu do specyficznych wymagań systemów robotycznych. Warto także rozważyć zastosowanie szybszych algorytmów optymalizacyjnych lub akceleracji obliczeń przy użyciu jednostek GPU, co mogłoby umożliwić stosowanie MPC w systemach o wysokiej dynamice w czasie rzeczywistym. Dodatkowo, porównanie metody MPC z innymi technikami identyfikacji, takimi jak filtry Kalmana czy estymacja adaptacyjna, w różnych scenariuszach dynamicznych, może dostarczyć cennych informacji na temat jej zalet i ograniczeń.

Podsumowując, przeprowadzone badania potwierdziły potencjał metody MPC w precyzyjnej identyfikacji parametrów manipulatorów, a wskazane kierunki dalszych badań stanowią podstawę do dalszego rozwoju tej technologii. Wyniki mogą znaleźć zastosowanie w robotyce przemysłowej, medycznej oraz w systemach autonomicznych, otwierając nowe możliwości w projektowaniu i sterowaniu manipulatorów o wysokiej precyzji.

Literatura

- [1] W. Domski, P. Bogdoł. Sterowanie predykcyjne oparte na modelu dla problemu sterowania rojem robotów. *Prace Naukowe Politechniki Warszawskiej. Elektronika*, 2022.
- [2] W. Domski, A. Mazur. Input-output decoupling for a 3D free-floating satellite with a 3R manipulator with state and input disturbances. *Bulletin of the Polish Academy of Sciences. Technical Sciences*, 2019.
- [3] J. M. Maciejowski. *Predictive Control with Constraints*. Prentice Hall, Harlow, UK, 2002.
- [4] J. B. Rawlings, D. Q. Mayne. *Model Predictive Control: Theory and Design*. Nob Hill Publishing, Madison, 2009.
- [5] K. Tchoń, A. Mazur, I. Duleba, R. Hossa, R. Muszyński. *Manipulatory i roboty mobilne: modele, planowanie ruchu, sterowanie*. PLJ, Warszawa, 2000.
- [6] M. Ławryńczuk. *Nonlinear Predictive Control Using Wiener Models*. Springer Cham, Cham, 2022.