

Politechnika Warszawska

WYDZIAŁ ELEKTRONIKI
I TECHNIK INFORMACYJNYCH



Instytut Automatyki i Informatyki Stosowanej

Praca dyplomowa inżynierska

na kierunku Automatyka i Robotyka

Efektywny układ stabilizacji odwróconego wahadła na wózku

Adam Sokołowski

Numer albumu 324892

promotor
mgr inż. Robert Nebeluk

WARSZAWA 2026

Efektywny układ stabilizacji odwróconego wahadła na wózku

Streszczenie.

Celem pracy było zaprojektowanie autorskiego środowiska symulacyjnego oraz przeprowadzenie wielokryterialnej analizy porównawczej algorytmów sterowania dla nieliniowego układu odwróconego wahadła na wózku. W ramach badań zaimplementowano i przetestowano cztery strategie sterowania: klasyczny regulator równoległy PID, układ hybrydowy PID-LQR, nieliniowe sterowanie predykcyjne (MPC) w dwóch wariantach funkcji kosztu oraz sterownik rozmyty typu Takagi-Sugeno wspomagany regulatorem LQR (Fuzzy-LQR). Badania przeprowadzono w środowisku języka Python, wykorzystując pełny model nieliniowy obiektu. Skuteczność algorytmów zweryfikowano w dwóch scenariuszach: stabilizacji w punkcie pracy oraz pracy w warunkach zakłóceń zewnętrznych (symulacja wiatru). Analiza wyników wykazała, że nie istnieje jeden uniwersalny regulator dla wszystkich zastosowań. Sterownik Fuzzy-LQR zapewnił najwyższą precyzję stabilizacji i odporność na zakłócenia, jednak odbyło się to kosztem bardzo wysokiego zużycia energii sterowania. Z kolei regulator MPC okazał się rozwiązaniem najbardziej ekonomicznym, zapewniając płynność ruchu i uwzględnienie ograniczeń fizycznych napędu, co jest kluczowe w systemach rzeczywistych. Układ PID-LQR stanowił kompromis między złożonością obliczeniową a jakością regulacji, przewyższając klasyczne podejście PID.

Słowa kluczowe: odwrócone wahadło, regulator PID, regulator LQR, regulator MPC, Takagi-Sugeno, sterowanie predykcyjne, sterowanie optymalne, zakłócenia, stabilizacja, symulacja, układ nieliniowy

Effective stabilisation system of the inverted pendulum on the cart

Abstract.

The aim of this thesis was to design a custom simulation environment and conduct a multi-criteria comparative analysis of control algorithms for a nonlinear inverted pendulum system on a cart. As part of the research, four control strategies were implemented and tested: a classical parallel PID controller, a hybrid PID-LQR system, nonlinear Model Predictive Control (MPC) with two cost function variants, and a Takagi-Sugeno fuzzy controller supported by LQR (Fuzzy-LQR). The study was conducted in a Python environment using a full nonlinear model of the object. The effectiveness of the algorithms was verified in two scenarios: stabilization at the operating point and operation under external disturbances (simulated wind). The analysis of the results showed that there is no single universal controller for all applications. The Fuzzy-LQR controller provided the highest stabilization precision and robustness to disturbances, but at the cost of very high control energy consumption. On the other hand, the MPC controller proved to be the most economical solution, ensuring smooth motion and compliance with physical drive constraints, which is crucial in real-world systems. The PID-LQR system represented a compromise between computational complexity and control quality, outperforming the classical PID approach.

Keywords: inverted pendulum, PID controller, LQR controller, MPC controller, Takagi-Sugeno, predictive control, optimal control, disturbances, stabilization, nonlinear system, simulation

Spis treści

1. Wstęp	7
1.1. Cel i zakres pracy	8
1.2. Przegląd literatury	9
1.3. Układ pracy	10
2. Model matematyczny układu	12
2.1. Opis fizyczny i założenia upraszczające	12
2.2. Analiza kinematyczna	13
2.3. Równania dynamiki układu	14
2.3.1. Bilans sił w kierunku poziomym	14
2.3.2. Bilans momentów sił	15
2.4. Nieliniowy model w przestrzeni stanów	15
2.5. Linearyzacja modelu w punkcie pracy	16
2.5.1. Analiza wartości własnych układu otwartego	17
2.5.2. Sterowność i obserwowalność	17
2.6. Analiza zachowania układu w pętli otwartej	18
3. Środowisko symulacyjne i implementacja	19
3.1. Narzędzia programistyczne	19
3.2. Konfiguracja symulacji	19
3.3. Modelowanie zakłóceń	20
3.4. Wizualizacja i animacja	21
4. Algorytmy sterowania	23
4.1. Równoległy regulator PID	23
4.1.1. Proces doboru nastaw oraz analiza PID	24
4.2. Układ hybrydowy PID-LQR	27
4.2.1. Dobór wag macierzy Q i R	29
4.3. Nieliniowe sterowanie predykcyjne (MPC)	31
4.3.1. Dobór horyzontu i wag funkcji celu	33
4.4. MPC z rozszerzonym wskaźnikiem jakości (MPC-J2)	35
4.4.1. Analiza wpływu kary za energię	36
4.5. Regulator rozmyty wspomagany LQR (Fuzzy-LQR)	39
4.5.1. Dobór reguł i funkcji przynależności	40
5. Eksperymenty	44
5.1. Plan eksperymentów	44
5.2. Badane algorytmy	45
5.3. Wskaźniki jakości regulacji	45
6. Analiza wyników	47
6.1. Stabilizacja w warunkach nominalnych	47

6.1.1. Charakterystyka regulatorów klasycznych	47
6.1.2. Charakterystyka regulatorów zaawansowanych	48
6.2. Analiza odporności na zakłócenia	50
6.3. Szczegółowe zestawienie ilościowe	52
6.4. Analiza wpływu kary za sterowanie w MPC z alternatywną funkcją kosztu . .	54
6.5. Porównanie złożoności obliczeniowej	55
7. Podsumowanie	57
7.1. Wnioski końcowe	57
7.2. Ograniczenia pracy	57
7.3. Kierunki dalszych badań	58
Bibliografia	59
Wykaz symboli i skrótów	61
Spis rysunków	62
Spis tabel	62

1. Wstęp

Odwrócone wahadło na wózku jest klasycznym przykładem nieliniowego, niestabilnego układu mechanicznego, wykorzystywanym powszechnie zarówno w dydaktyce, jak i w badaniach nad zaawansowanymi technikami sterowania. Mimo że geometria i parametry fizyczne obiektu są stosunkowo proste, układ ten pozostaje wyzwaniem w zakresie stabilizacji i wymaga precyzyjnej regulacji w czasie rzeczywistym. Jego charakterystyczna cecha — podwzbudność (ang. *underactuated system*), oznaczająca mniejszą liczbę wejść sterujących niż wyjść — oraz silna wrażliwość na zakłócenia sprawiają, że nawet niewielkie odchylenia mogą prowadzić do gwałtownego narastania błędów i utraty równowagi.

Znaczenie tego modelu wykracza daleko poza cele czysto akademickie. Odwrócone wahadło na wózku służy jako kanoniczny *benchmark* dla metod sterowania i estymacji stanu, ponieważ łączy w sobie trudności typowe dla systemów rzeczywistych: nieliniowość, niestabilność w otwartym układzie sterowania, ograniczenia aktuatora oraz niepewność parametrów. Umożliwia to weryfikację algorytmów w sytuacjach, w których klasyczne założenia teorii liniowej przestają obowiązywać, a układ wymaga adaptacji lub podejścia optymalnego.

Model ten posiada liczne analogie w praktyce inżynierskiej. Jego dynamika odwzorowuje wiele złożonych zjawisk fizycznych i konstrukcji technicznych, w tym:

- stabilizację robotów dwukołowych (np. typu Segway) oraz robotów mobilnych balansujących na jednej osi [1];
- sterowanie raketami nośnymi i pociskami w fazie startu, gdzie wektor ciągu pełni rolę siły napędowej wózka, a bezwładność korpusu odpowiada dynamice wahadła [2];
- sterowanie ruchem ramion manipulatorów i suwnic, w których kluczowe jest tłumienie oscylacji przenoszonego ładunku [3];
- równoważenie platform i pojazdów samobalansujących, wymagające ciągłej korekty siły napędowej względem położenia środka masy [4].

Ze względu na powyższe zastosowania, problem stabilizacji odwróconego wahadła traktowany jest jako uproszczony model systemów rzeczywistych o zbliżonej dynamice. Badania symulacyjne na tym obiekcie pozwalają na wstępną walidację skuteczności algorytmów sterowania przed ich implementacją w bardziej złożonych lub kosztownych systemach. Z tego powodu układ ten od dziesięcioleci stanowi punkt odniesienia w rozwoju nowoczesnych metod regulacji — od klasycznych regulatorów PID (regulator proporcjonalno-całkująco-różniczkujący) i LQR (liniowy regulator kwadratowy), po sterowanie predykcyjne (MPC), adaptacyjne i rozmyte.

Prostota modelu matematycznego w połączeniu z łatwością interpretacji wyników (analiza kąta wychylenia i pozycji wózka) sprawiają, że odwrócone wahadło łączy elegancję

analityczną z praktycznymi wyzwaniami inżynierskimi. Stanowi tym samym uniwersalne narzędzie do nauki, testowania i rozwijania metod stabilizacji systemów nieliniowych.

1.1. Cel i zakres pracy

Zasadniczym celem pracy jest zaprojektowanie i realizacja autorskiego środowiska symulacyjnego oraz przeprowadzenie w nim wielokryterialnej analizy porównawczej algorytmów sterowania dla nieliniowego układu odwróconego wahadła.

Aby zrealizować ten cel, zdefiniowano szereg wymagań funkcjonalnych i нефunkcjonalnych stawianych opracowywanemu systemowi. W warstwie funkcjonalnej środowisko musi wiernie odwzorowywać dynamikę układu wózka z wahadłem w oparciu o nieliniowe równania ruchu, a także umożliwiać elastyczne przełączanie pomiędzy zaimplementowanymi strategiami sterowania (PID, LQR, MPC, Fuzzy). Kluczowa jest również możliwość weryfikacji odporności regulatorów poprzez generowanie addytywnych zakłóceń zewnętrznych (np. siły wiatru). System powinien oferować narzędzia do wizualizacji przebiegów oraz animacji ruchu obiektu, a także automatycznie archiwizować wyniki eksperymentów w celu późniejszego wyznaczenia wskaźników jakości (MSE, MAE, energia sterowania).

W zakresie wymagań нефunkcjonalnych przyjęto, że oprogramowanie zostanie zrealizowane w języku Python 3 z wykorzystaniem bibliotek numerycznych NumPy i SciPy. Projekt musi charakteryzować się wysoką modularnością, separującą logikę sterowania od silnika symulacyjnego, co ułatwi jego dalszą rozbudowę. Ponadto, zaimplementowane algorytmy powinny cechować się wydajnością umożliwiającą pracę w czasie zbliżonym do rzeczywistego.

Większość dostępnych publikacji skupia się zazwyczaj tylko na jednej grupie metod. W niniejszej pracy postanowiono porównać ze sobą trzy zupełnie różne podejścia:

- klasyczne (PID, kaskady),
- optymalne i predykcyjne (LQR, MPC),
- sterowanie rozmyte (Fuzzy).

Wszystkie te algorytmy zostały uruchomione w tym samym środowisku symulacyjnym. Dzięki temu możliwe było ich uczciwe porównanie w identycznych warunkach. Sprawdzono nie tylko, jak precyzyjnie utrzymują wahadło w pionie, ale także jak radzą sobie z zakłóceniami (np. porywistym wiatrem) oraz ile energii zużywają podczas pracy.

W ramach projektu zaimplementowano i poddano badaniom następujące struktury:

1. **Regulator PID** — reprezentujący klasyczne podejście inżynierskie. Układ składa się z dwóch pętli sprzężenia zwrotnego (stabilizacja kąta i pozycji), wyposażonych w mechanizmy zapobiegające nasyceniu członów całkujących (ang. *anti-windup*) oraz ograniczenia sygnału sterującego.
2. **Hybrydowy układ PID-LQR** — struktura łącząca prostotę PID (w pętli regulacji pozycji) z optymalnym regulatorem stanu LQR (w pętli stabilizacji wahadła). Metoda

ta stanowi punkt odniesienia dla oceny skuteczności metod bazujących na modelu liniowym.

3. **Regulator MPC** — wariant podstawowy sterowania predykcyjnego z kwadratową funkcją kosztu. Służy on w pracy jako nowoczesny *benchmark*, pozwalający ocenić, jakie korzyści daje uwzględnienie ograniczeń sterowania i dynamiki układu bezpośrednio w procesie optymalizacji.
4. **Regulator MPC z rozszerzonym wskaźnikiem jakości** — wariant badawczy metody predykcyjnej, w którym przeanalizowano wpływ dodatkowych kar w funkcji celu (za gwałtowne zmiany sterowania oraz jego amplitudę) na płynność regulacji i zużycie energii.
5. **Regulator rozmyty (Fuzzy Logic)** — metoda sterowania inteligentnego, wykorzystująca zbiór reguł wnioskowania, stanowiąca alternatywę dla metod analitycznych.

Kluczowym elementem pracy jest weryfikacja działania regulatorów w zadaniu stabilizacji wahadła w pozycji pionowej (punkt pracy) w obecności zakłóceń zewnętrznych. W celu obiektywnego porównania metod przyjęto zestaw wskaźników ilościowych. Jakość stabilizacji weryfikowana jest w oparciu o metryki całkowite błędu (ISE, IAE) dla kąta wychylenia oraz pozycji wózka. Równolegle ocenie poddano ekonomię sterowania, wyznaczając koszt energetyczny poprzez normy L_1 i L_2 sygnału sterującego. Całość uzupełnia analiza odporności układu na zakłócenia (szumy, siły zewnętrzne). Tak dobrane kryteria pozwalają na wszechstronne wskazanie mocnych i słabych stron badanych metod.

1.2. Przegląd literatury

W literaturze przedmiotu odwrócone wahadło na wózku traktowane jest powszechnie jako wzorcowy układ testowy (ang. *testbed*) dla weryfikacji algorytmów sterowania układami niestabilnymi. W pracy [5] przedstawiono kompletny model nieliniowy obiektu, uwzględniający zakłócenia stochastyczne, na podstawie którego przeprowadzono analizę porównawczą regulatorów PID, LQR oraz ich konfiguracji hybrydowych. Wyniki te wskazują, że włączenie komponentu LQR znacząco poprawia szybkość i płynność odpowiedzi w stosunku do klasycznej regulacji PID, szczególnie w zakresie stabilizacji kątowej wahadła.

Rozszerzenie zakresu badań o sterowanie predykcyjne (MPC) zaprezentowano w pozycji [6]. Autorzy stworzyli jednorodne środowisko symulacyjne, zestawiając przebiegi zmiennych stanu dla metod PID, LQR oraz MPC. Uzyskane rezultaty potwierdziły przewagę rozwiązań opartych na modelu (LQR, MPC) nad klasycznym PID w kontekście jakości regulacji, podkreślając jednocześnie kluczową zaletę MPC — możliwość bezpośredniego uwzględniania ograniczeń fizycznych nałożonych na wielkości sterujące.

Współczesne prace badawcze coraz częściej integrują metody optymalne z metodami sztucznej inteligencji. Artykuł [7] opisuje rozwiązanie hybrydowe, łączące regulator LQR z modelem rozmytym Takagi–Sugeno (z kompensacją PDC) oraz obserwatorem stanu. Podejście to pozwala na przyspieszenie zbieżności błędu regulacji do zera oraz poprawę

jakości estymacji zmiennych w obecności szumów pomiarowych i niepewności parametrycznej modelu.

Istotnym uzupełnieniem badań symulacyjnych są weryfikacje eksperymentalne, szeroko reprezentowane w krajowej literaturze naukowej. Zespół Jezierski i in. [8] przeprowadził porównanie algorytmów LQR i MPC na rzeczywistym stanowisku laboratoryjnym. Wykazano, że o ile regulator LQR skutecznie utrzymuje punkt pracy i tłumi zakłócenia, to sterowanie predykcyjne zapewnia łagodniejsze sterowanie i lepsze właściwości śledzenia trajektorii, co ma kluczowe znaczenie w aplikacjach robotycznych.

Z punktu widzenia podstaw teoretycznych, fundamentem dla implementacji sterowania predykcyjnego są prace monograficzne Camacho i Bordonsa [9] oraz Tatjewskiego [10]. Omawiają one szczegółowo zagadnienia doboru funkcji kosztu, horyzontów predykcji, a także stabilności układu zamkniętego. Aspekty wdrożeniowe, w tym efektywność numeryczna algorytmów optymalizacji na platformach wbudowanych, poruszane są w nowszych publikacjach [11], [12]. Natomiast w obszarze sterowania rozmytego cennym źródłem wiedzy metodycznej są opracowania dotyczące modeli Takagi–Sugeno i ich porównań z podejściami klasycznymi [13].

Przeprowadzona analiza literatury wskazuje na ewolucję podejść sterowania: od klasycznych paradygmatów PID i LQR [5], przez ujęcia predykcyjne [6], [10], aż po zaawansowane metody hybrydowe i inteligentne [7].

1.3. Układ pracy

Układ pracy został podzielony na rozdziały, których treść odpowiada kolejnym etapom realizacji projektu.

Rozdział drugi przedstawia szczegółowe wyprowadzenie modelu matematycznego odwróconego wahadła na wózku. Zawiera on opis fizyczny obiektu, równania dynamiki sformułowane w oparciu o prawa mechaniki, a także linearyzację modelu niezbędną do syntezy wybranych regulatorów.

Rozdział trzeci poświęcony jest opisowi zrealizowanego środowiska symulacyjnego. Przedstawiono w nim narzędzia programistyczne, metody numeryczne wykorzystane do rozwiązywania równań różniczkowych oraz sposób modelowania zakłóceń zewnętrznych.

Rozdział czwarty zawiera charakterystykę zaimplementowanych algorytmów sterowania. Omówiono w nim podstawy teoretyczne oraz szczegóły implementacyjne regulatorów: klasycznego PID, optymalnego LQR, predykcyjnego MPC oraz rozmytego Takagi–Sugeno.

Rozdział piąty opisuje metodykę badań symulacyjnych. Zdefiniowano w nim scenariusze testowe, przyjęte wskaźniki jakości oraz procedurę strojenia regulatorów, ze szczególnym uwzględnieniem doboru wag macierzy LQR oraz nastaw członu PID.

Rozdział szósty prezentuje wyniki przeprowadzonych eksperymentów. Zawiera on szczegółową analizę przebiegów czasowych, zestawienie tabelaryczne błędów regulacji

w warunkach nominalnych i zakłóconych, oraz dyskusję porównawczą skuteczności badanych metod.

Pracę kończy podsumowanie, zawierające wnioski końcowe oraz kierunki dalszego rozwoju projektu.

2. Model matematyczny układu

Celem niniejszego rozdziału jest szczegółowe wyprowadzenie modelu matematycznego obiektu sterowania, którym jest odwrócone wahadło na wózku. Precyzyjne odwzorowanie dynamiki procesu jest fundamentalnym etapem projektowania układu sterowania, gdyż jakość modelu bezpośrednio wpływa na skuteczność algorytmów predykcyjnych (MPC) oraz optymalnych (LQR). Wyprowadzenie oparto na prawach mechaniki klasycznej oraz analizie sił i momentów przedstawionej w literaturze przedmiotu [5].

2.1. Opis fizyczny i założenia upraszczające

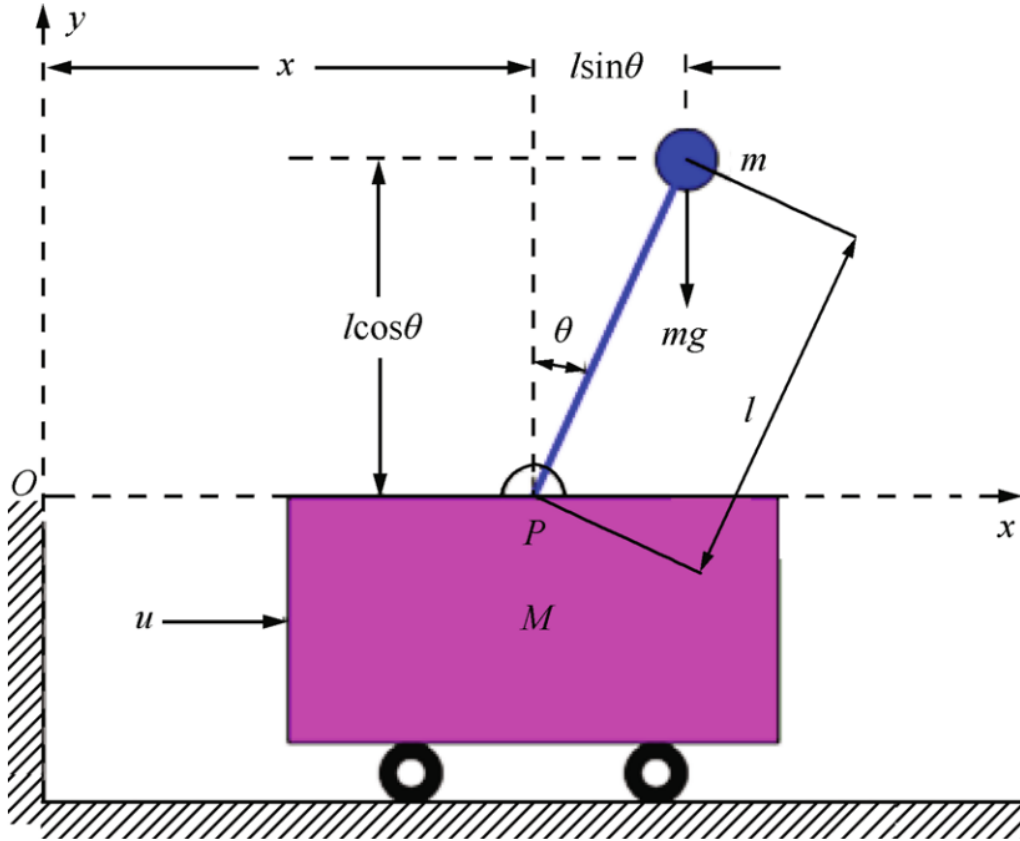
Rozważany obiekt sterowania należy do klasy mechanicznych układów podwzbudzonych (ang. *underactuated systems*), co oznacza, że liczba wejść sterujących jest mniejsza od liczby stopni swobody. Układ składa się z wózka poruszającego się wzdłuż poziomej osi oraz pręta z masą skupioną, zamocowanego przegubowo do wózka.

Przyjęto następujące parametry fizyczne modelu, zgodne z oznaczeniami stosowanymi w pracach badawczych [5]:

- M — masa wózka [kg],
- m — masa wahadła (traktowana jako masa punktowa na końcu pręta) [kg],
- l — długość wahadła (odległość od osi obrotu do środka ciężkości) [m],
- g — przyspieszenie ziemskie [m/s^2],
- $F(t)$ — siła sterująca przyłożona do wózka [N],
- $F_w(t)$ — siła zakłócająca (wiatr) działająca poziomo na masę wahadła [N].

W celu sformułowania modelu analitycznego przyjęto następujące założenia upraszczające [5]:

1. Pręt wahadła jest nieważki i sztywny.
2. Tarcie w łożyskach kół wózka oraz w przegubie wahadła jest pomijalnie małe.
3. Ruch odbywa się w płaszczyźnie pionowej $x - y$.
4. Cała masa wahadła jest skupiona w jego środku geometrycznym (masa punktowa).



Rysunek 2.1. Źródło: [5]. Schemat układu odwróconego wahadła na wózku.

Układ odniesienia zdefiniowano w taki sposób, że współrzędna $x(t)$ opisuje poziome przesunięcie wózka, natomiast kąt $\theta(t)$ określa wychylenie wahadła od pionu w kierunku zgodnym z ruchem wskazówek zegara. Pozycja $\theta = 0$ odpowiada niestabilnemu punktowi równowagi (wahadło w górze).

2.2. Analiza kinematyczna

W pierwszej kolejności zdefiniowano położenie środka ciężkości (COG) masy wahadła m w nieruchomym układzie odniesienia. Oznaczając współrzędne środka masy jako (x_G, y_G) , można je wyrazić jako sumę przemieszczenia wózka oraz rzutu geometrycznego ramienia wahadła [5]:

$$x_G(t) = x(t) + l \sin \theta(t), \quad (1)$$

$$y_G(t) = l \cos \theta(t). \quad (2)$$

Gdzie x_G to pozioma współrzędna masy wahadła, a y_G to jej współrzędna pionowa (odległość od osi wózka).

2. Model matematyczny układu

Aby zastosować drugą zasadę dynamiki Newtona dla masy m , konieczne jest wyznaczenie jej prędkości oraz przyspieszeń całkowitych (bezwzględnych). Różniczkując równania (1) i (2) względem czasu, otrzymuje się składowe prędkości:

$$\dot{x}_G(t) = \dot{x}(t) + l \cos \theta(t) \cdot \dot{\theta}(t), \quad (3)$$

$$\dot{y}_G(t) = -l \sin \theta(t) \cdot \dot{\theta}(t). \quad (4)$$

Ponowne różniczkowanie pozwala wyznaczyć składowe przyspieszenia środka masy wahadła \ddot{x}_G oraz \ddot{y}_G . Uwzględniają one zarówno przyspieszenie liniowe wózka, jak i składowe ruchu obrotowego (przyspieszenie styczne i dośrodkowe):

$$\ddot{x}_G(t) = \ddot{x}(t) + l \cos \theta(t) \cdot \ddot{\theta}(t) - l \sin \theta(t) \cdot \dot{\theta}^2(t), \quad (5)$$

$$\ddot{y}_G(t) = -l \sin \theta(t) \cdot \ddot{\theta}(t) - l \cos \theta(t) \cdot \dot{\theta}^2(t). \quad (6)$$

Powyższe zależności (5) i (6) są kluczowe, ponieważ siły bezwładności działające na masę wahadła zależą od jej całkowitego przyspieszenia w przestrzeni, a nie tylko od kąta wychylenia.

2.3. Równania dynamiki układu

Model dynamiczny wyprowadzono, rozpatrując siły działające na wózek oraz na wahadło osobno, a następnie składając je w układ równań sprzężonych.

2.3.1. Bilans sił w kierunku poziomym

Zgodnie z drugą zasadą dynamiki Newtona, suma sił zewnętrznych działających na cały układ w kierunku osi x musi równać się zmianie pędu układu. Siłami zewnętrznymi są: siła sterująca $u(t)$ przyłożona do wózka oraz siła zakłócająca $F_w(t)$ przyłożona do masy wahadła [5].

Równanie równowagi sił dla całego układu (masy M i m) przyjmuje postać:

$$M\ddot{x}(t) + m\ddot{x}_G(t) = u(t) + F_w(t). \quad (7)$$

Podstawiając wyznaczone wcześniej przyspieszenie poziome środka masy wahadła (5) do równania (7), otrzymujemy:

$$M\ddot{x}(t) + m(\ddot{x}(t) + l \cos \theta(t) \cdot \ddot{\theta}(t) - l \sin \theta(t) \cdot \dot{\theta}^2(t)) = u(t) + F_w(t). \quad (8)$$

Po uporządkowaniu wyrazów i wyciągnięciu \ddot{x} przed nawias, otrzymujemy pierwsze równanie różniczkowe opisujące ruch postępowy układu:

$$(M + m)\ddot{x}(t) + ml \cos \theta(t) \cdot \ddot{\theta}(t) - ml \sin \theta(t) \cdot \dot{\theta}^2(t) = u(t) + F_w(t). \quad (9)$$

2.3.2. Bilans momentów sił

Ruch obrotowy wahadła opisano poprzez sumę momentów sił względem punktu zawieszenia (przegubu na wózku). Siłami generującymi moment obrotowy są siły bezwładności masy wahadła, siła ciężkości oraz siła zakłócająca.

Zgodnie z analizą przedstawioną w pracy [5], równanie momentów przyjmuje postać:

$$m\ddot{x}_G(t)l\cos\theta(t) - m\ddot{y}_G(t)l\sin\theta(t) = mgl\sin\theta(t) + F_w(t)l\cos\theta(t). \quad (10)$$

W równaniu tym lewa strona reprezentuje moment wynikający z sił bezwładności, natomiast prawa strona uwzględnia momenty od sił zewnętrznych (grawitacji i wiatru). Należy zauważyć, że siła wiatru $F_w(t)$ działa poziomo, stąd jej ramię siły względem punktu obrotu wynosi $l\cos\theta(t)$.

Podstawiając wyrażenia na przyspieszenia \ddot{x}_G (5) oraz \ddot{y}_G (6) do równania momentów (10), otrzymujemy rozbudowaną postać równania:

$$\begin{aligned} & ml\cos\theta(t)(\ddot{x}(t) + l\cos\theta(t)\ddot{\theta}(t) - l\sin\theta(t)\dot{\theta}^2(t)) \\ & - ml\sin\theta(t)(-l\sin\theta(t)\ddot{\theta}(t) - l\cos\theta(t)\dot{\theta}^2(t)) = mgl\sin\theta(t) + F_w(t)l\cos\theta(t). \end{aligned} \quad (11)$$

Po wymnożeniu nawiasów składniki zawierające $\dot{\theta}^2$ wzajemnie się redukują. Wykorzystując jedynkę trygonometryczną $\cos^2\theta + \sin^2\theta = 1$ oraz dzieląc całe równanie przez l , otrzymujemy ostateczne drugie równanie dynamiki [5]:

$$m\cos\theta(t) \cdot \ddot{x}(t) + ml \cdot \ddot{\theta}(t) = mg\sin\theta(t) + F_w(t)\cos\theta(t). \quad (12)$$

2.4. Nieliniowy model w przestrzeni stanów

Układ równań (9) i (12) stanowi kompletny, sprzężony opis dynamiki. W celu przeprowadzenia symulacji numerycznej, konieczne jest rozprężenie układu i wyznaczenie jawnych postaci przyspieszeń \ddot{x} i $\ddot{\theta}$.

Wyznaczając \ddot{x} z równania (12):

$$\ddot{x}(t) = \frac{1}{m\cos\theta(t)} (mg\sin\theta(t) + F_w(t)\cos\theta(t) - ml\ddot{\theta}(t)), \quad (13)$$

i podstawiając do równania (9), a następnie wykonując przekształcenia algebraiczne, otrzymujemy jawne wzory na przyspieszenia.

Ostateczne równania ruchu, uwzględniające wpływ siły sterującej u oraz zakłócenia F_w , przyjmują postać [5]:

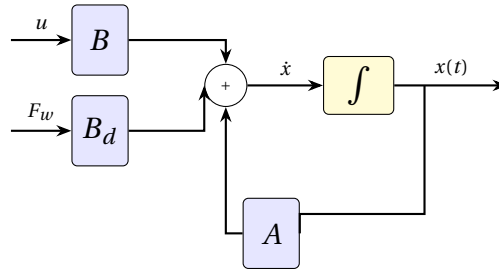
$$\ddot{x}(t) = \frac{u(t) + ml\sin\theta(t) \cdot \dot{\theta}^2(t) - mg\sin\theta(t)\cos\theta(t) + F_w(t)\sin^2\theta(t)}{M + m - m\cos^2\theta(t)}, \quad (14)$$

$$\ddot{\theta}(t) = \frac{(M+m)g \sin \theta(t) - u(t) \cos \theta(t) - ml \sin \theta(t) \cos \theta(t) \cdot \dot{\theta}^2(t) - \frac{M}{m} F_w(t) \cos \theta(t) + F_w(t) \cos \theta(t)}{l(M+m-m \cos^2 \theta(t))}. \quad (15)$$

Definiując wektor stanu $\mathbf{x} = [\theta, \dot{\theta}, x, \dot{x}]^T$, model w przestrzeni stanów zapisujemy jako układ czterech równań różniczkowych pierwszego rzędu:

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \\ \dot{x}_4 \end{bmatrix} = \begin{bmatrix} x_2 \\ \frac{(M+m)g \sin x_1 - u \cos x_1 - ml(\sin x_1 \cos x_1)x_2^2 - F_w(\frac{M}{m} \cos x_1 - \cos x_1)}{l(M+m-m \cos^2 x_1)} \\ x_4 \\ \frac{u + ml(\sin x_1)x_2^2 - mg \sin x_1 \cos x_1 + F_w \sin^2 x_1}{M+m-m \cos^2 x_1} \end{bmatrix}. \quad (16)$$

Schemat blokowy modelu w przestrzeni stanów, ilustrujący przepływ sygnałów sterowania i zakłóceń, przedstawiono na Rys. 2.2.



Rysunek 2.2. Schemat blokowy nieliniowego modelu wahadła w przestrzeni stanów.

2.5. Linearyzacja modelu w punkcie pracy

W celu zastosowania algorytmów sterowania liniowego (PID, LQR), przeprowadzono linearyzację modelu nieliniowego wokół punktu równowagi chwiejnej ($\theta = 0$). Dla małych wychyleń zastosowano przybliżenia $\sin \theta \approx \theta$, $\cos \theta \approx 1$, a wyrazy wyższego rzędu ($\dot{\theta}^2$, $\sin^2 \theta$) pominięto.

Liniowy model układu w postaci $\dot{\mathbf{x}} = \mathbf{A}\mathbf{x} + \mathbf{B}u + \mathbf{B}_{\text{dist}}F_w$ opisują macierze wyznaczone zgodnie z literaturą [5]:

$$\mathbf{A} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ \frac{(M+m)g}{Ml} & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ -\frac{mg}{M} & 0 & 0 & 0 \end{bmatrix}. \quad (17)$$

Macierz sterowania B oraz macierz zakłóceń B_{dist} przyjmują postać:

$$B = \begin{bmatrix} 0 \\ -\frac{1}{Ml} \\ 0 \\ \frac{1}{M} \end{bmatrix}, \quad B_{\text{dist}} = \begin{bmatrix} 0 \\ -\frac{1}{ml} \\ 0 \\ 0 \end{bmatrix}. \quad (18)$$

Warto zauważyć, że w modelu zlinearyzowanym wpływ wiatru na przyspieszenie kątowe wahadła jest odwrotnie proporcjonalny do masy m (element $-\frac{1}{ml}$ w macierzy B_{dist}), podczas gdy w równaniu przyspieszenia wózka efekt ten zanika dla małych kątów (element zerowy). Jest to zgodne z wynikami prezentowanymi w pracy źródłowej [5].

2.5.1. Analiza wartości własnych układu otwartego

Aby formalnie potwierdzić niestabilność układu w górnym punkcie równowagi, wyznaczono wartości własne macierzy stanu A . Dla przyjętych parametrów fizycznych ($M = 2,4$ kg, $m = 0,23$ kg, $l = 0,36$ m, $g = 9,81$ m/s²) wielomian charakterystyczny $\det(\lambda I - A) = 0$ przyjmuje postać:

$$\lambda^4 - \frac{(M+m)g}{Ml} \lambda^2 = 0. \quad (19)$$

Rozwiązując powyższe równanie, otrzymujemy cztery wartości własne:

$$\lambda_{1,2} = 0, \quad \lambda_3 \approx -5,49, \quad \lambda_4 \approx +5,49. \quad (20)$$

Obecność dodatniej wartości własnej $\lambda_4 > 0$ potwierdza, że układ w pętli otwartej jest **niestabilny** — każde, nawet minimalne odchylenie od pionu będzie narastać eksponencjalnie w czasie. Dwie zerowe wartości własne odpowiadają ruchowi wózka po torze (brak tłumienia, brak siły przywracającej).

2.5.2. Sterowność i obserwowalność

Warunkiem koniecznym stosowalności regulatora LQR jest pełna sterowność pary (A, B) . Macierz sterowności Kalmana zdefiniowana jest jako:

$$\mathcal{C} = \begin{bmatrix} B & AB & A^2B & A^3B \end{bmatrix}. \quad (21)$$

Dla rozpatrywanego układu wyznaczono $\text{rank}(\mathcal{C}) = 4$, co oznacza **pełną sterowność** — istnieje sygnał sterujący $u(t)$, który pozwala przeprowadzić układ z dowolnego stanu początkowego do dowolnego stanu końcowego w skończonym czasie.

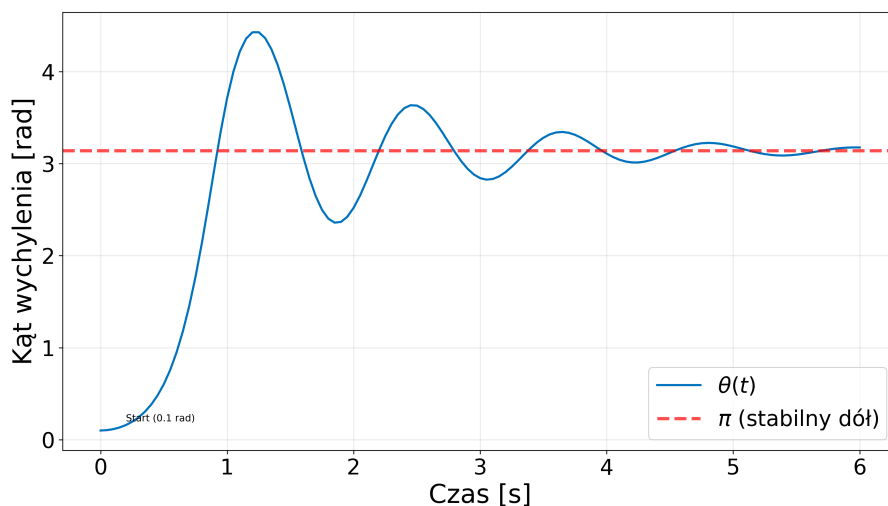
Analogicznie, macierz obserwowalności (przy założeniu pomiaru kąta θ i pozycji x , tj. $C = [1, 0, 0, 0; 0, 0, 1, 0]$) ma pełny rząd, co potwierdza **pełną obserwowalność** układu. Oznacza to, że na podstawie pomiarów wyjściowych możliwe jest jednoznaczne odtwo-

rzenie pełnego wektora stanu, co jest istotne dla praktycznej implementacji regulatorów opartych na sprzężeniu zwrotnym od stanu.

2.6. Analiza zachowania układu w pętli otwartej

Odwrócone wahadło jest z natury układem niestabilnym w górnym punkcie równowagi ($\theta = 0$). Aby zobrazować tę właściwość, przeprowadzono symulację zachowania obiektu bez działania układu sterowania ($u(t) = 0$), przy niewielkim wychyleniu początkowym $\theta_0 = 0,1$ rad (ok. $5,7^\circ$).

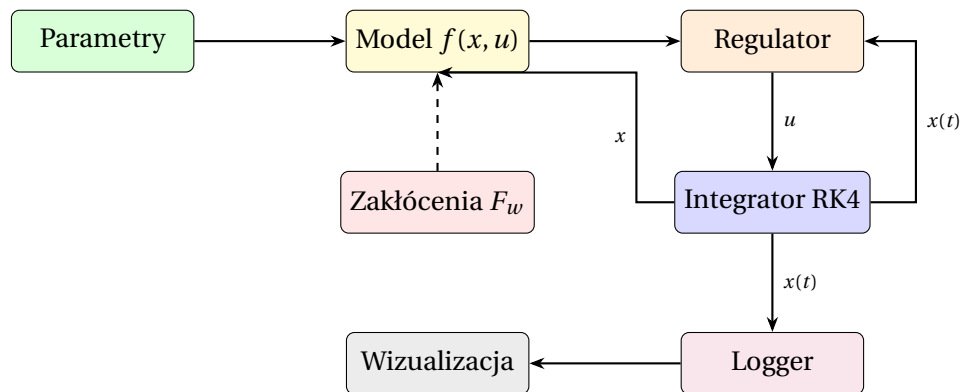
Jak pokazano na Rys. 2.3, nawet niewielkie odchylenie od pionu powoduje, że pod wpływem siły grawitacji wahadło gwałtownie traci równowagę i opada. W rzeczywistym układzie fizycznym, obecność tarcia w przegubie oraz oporu powietrza sprawia, że energia mechaniczna jest rozpraszana. W rezultacie, kąt wychylenia $\theta(t)$ dąży do stabilnego punktu równowagi dolnej, tj. $\theta = \pi$ rad.



Rysunek 2.3. Symulacja odpowiedzi swobodnej układu (z uwzględnionym tłumieniem) na małe wychylenie początkowe. Układ opuszcza niestabilny punkt równowagi ($\theta \approx 0$) i stabilizuje się w pozycji wiszącej ($\theta = \pi$).

3. Środowisko symulacyjne i implementacja

W celu przeprowadzenia badań i weryfikacji działania algorytmów sterowania, przygotowano autorskie środowisko symulacyjne zrealizowane w języku **Python 3**. Wybór tego języka podyktowany był jego powszechnością w zastosowaniach naukowych, dostępnością bibliotek do obliczeń numerycznych i optymalizacji, a także łatwością prototypowania złożonych struktur sterowania. Ogólną architekturę środowiska przedstawiono na Rys. 3.1.



Rysunek 3.1. Architektura środowiska symulacyjnego.

3.1. Narzędzia programistyczne

W projekcie wykorzystano następujące biblioteki i narzędzia:

- **NumPy** – podstawowa biblioteka do obliczeń macierzowych i operacji na wielowymiarowych tablicach danych, wykorzystywana do implementacji równań stanu oraz przechowywania przebiegów symulacji.
- **SciPy** – pakiet naukowy dostarczający zaawansowanych algorytmów numerycznych. W pracy użyto modułów:
 - `scipy.linalg` – do rozwiązywania algebraicznego równania Riccatiego (ARE) w algorytmie LQR.
 - `scipy.optimize` – zawierającego solver `minimize` (metoda SLSQP), wykorzystywany do rozwiązywania zadań optymalizacji nieliniowej z ograniczeniami w regulatorze MPC.
- **Matplotlib** – biblioteka służąca do wizualizacji wyników w postaci wykresów przebiegów czasowych oraz do generowania animacji ruchu wahadła.

3.2. Konfiguracja symulacji

Symulator opiera się na numerycznym całkowaniu wyprowadzonych wcześniej nieliniowych równań dynamiki. Zaimplementowano procedurę całkowania metodą Rungego-Kutty czwartego rzędu (RK4). Wybór tej metody podyktowany był kilkoma czynnikami: w odróżnieniu od prostszej metody Eulera, RK4 charakteryzuje się błędem lokalnym rzędu

$\mathcal{O}(\Delta t^5)$, co zapewnia wysoką dokładność przy umiarkowanym koszcie obliczeniowym. Jednocześnie, w przeciwieństwie do metod adaptacyjnych (np. ode45), stały krok czasowy gwarantuje deterministyczne taktowanie pętli sterowania, co jest istotne przy porównywaniu regulatorów. Przyjęto stały krok symulacji oraz sterowania wynoszący $\Delta t = 0,01$ s (10 ms).

Tabela 3.1. Parametry fizyczne modelu przyjęte w symulacji

Parametr	Symbol	Wartość	Jednostka
Masa wózka	M	2,40	kg
Masa wahadła	m	0,23	kg
Długość wahadła	l	0,36	m
Przyspieszenie ziemskie	g	9,81	m/s ²
Ograniczenie sterowania	F_{\max}	100,00	N

Symulacje przeprowadzane są dla zadania stabilizacji układu w pionie (tzw. punkt pracy), startując z niezerowych warunków początkowych lub wymuszając zmianę pozycji wózka.

Warunki początkowe (domyślne):

$$\mathbf{x}_0 = [\theta, \dot{\theta}, x, \dot{x}]^T = [0,05, 0, 0, 0]^T$$

Oznacza to niewielkie (ok. 2,86°) początkowe wychylenie wahadła, które regulator musi zniwelować.

Wartości zadane: Celem układu jest osiągnięcie stanu $\mathbf{x}_{\text{ref}} = [0, 0, x_{\text{ref}}, 0]^T$, gdzie x_{ref} (np. 0,10 m) jest zadaną nową pozycją wózka, przy jednoczesnym utrzymaniu pionowej pozycji wahadła ($\theta = 0$).

3.3. Modelowanie zakłóceń

Aby zweryfikować odporność układów sterowania, zaimplementowano generator zakłóceń symulujący podmuchy wiatru działające na wahadło. Generator ten działa w sposób dyskretny, realizując w każdym kroku symulacji k następujące operacje:

1. Dyskretnie próbkowanie szumu:

$$w_k \sim \mathcal{N}(0, \sigma^2), \quad (22)$$

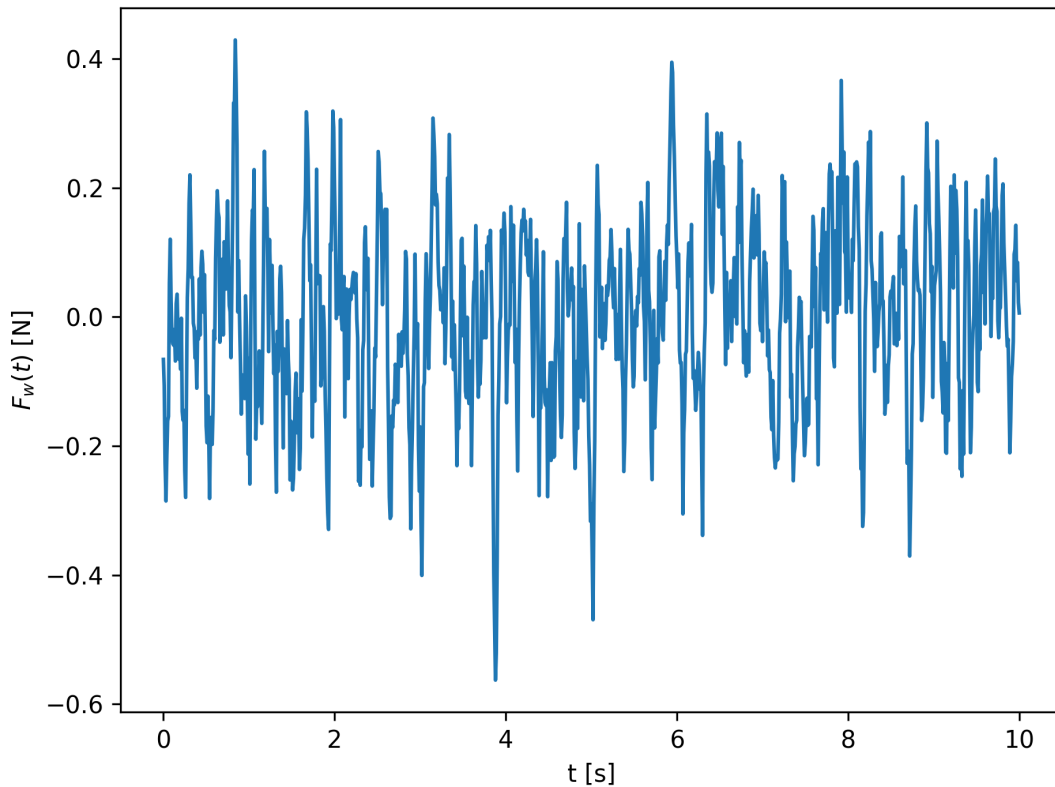
gdzie σ jest odchyleniem standardowym siły zakłócającej wyrażonym w niutonach. W przeprowadzonych eksperymentach przyjęto $\sigma = 2,2$ N, co oznacza, że wartość skuteczna (RMS) siły wiatru wynosi około 2,2 N, a chwilowe wartości szczytowe mogą osiągać $\pm 6,6$ N (przedział 3σ). Parametr ten dobrano empirycznie tak, aby zakłócenia stanowiły istotne

wyzwanie dla układu sterowania (porównywalne z kilkoma procentami maksymalnej siły aktuatora $F_{\max} = 100$ N), lecz nie przekraczały możliwości kompensacyjnych badanych regulatorów.

2. Wygładzanie (ruchoma średnia):

$$F_{w,k} = \frac{1}{N_s} \sum_{i=0}^{N_s-1} w_{k-i}, \quad (23)$$

gdzie $F_{w,k}$ to wypadkowa siła wiatru w chwili k , a N_s to długość okna uśredniającego (przyjęto $N_s = 10$). Takie podejście pozwala na uzyskanie ciągłego, wolnozmiennego sygnału lepiej odwzorowującego rzeczywiste podmuchy. Przykładowy przebieg wygenerowanego sygnału przedstawiono na Rys. 3.2.



Rysunek 3.2. Przykładowa realizacja stochastycznego procesu zakłócenia (wiatru) działającego na wahadło w czasie symulacji.

3.4. Wizualizacja i animacja

Oprócz standardowych wykresów zmiennych stanu i sterowania, środowisko wyposażono w moduł wizualizacji dynamicznej (Rys. 3.3). Implementacja animacji oparta jest na bibliotece Matplotlib i klasie FuncAnimation, która pozwala na cykliczne odświeżanie obiektów graficznych zgodnie z taktowaniem symulacji.

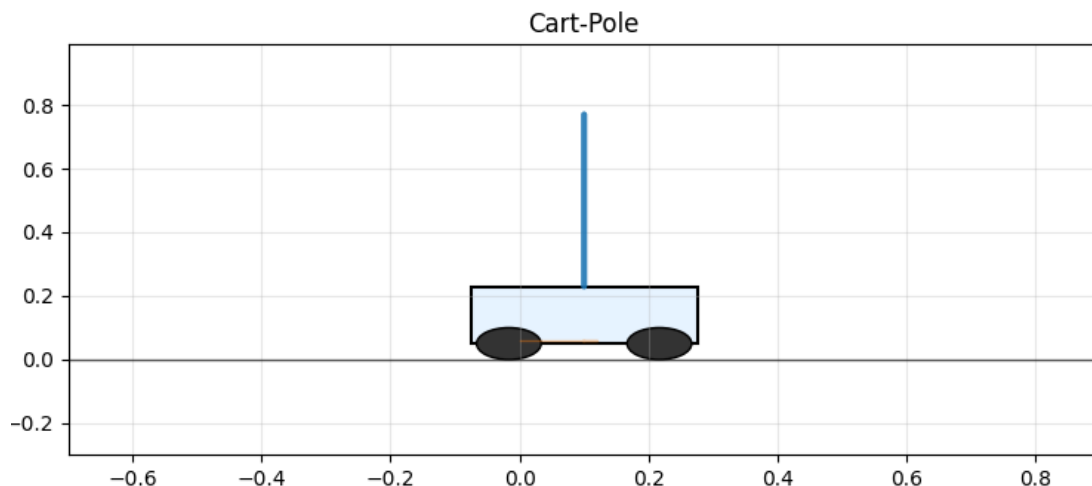
3. Środowisko symulacyjne i implementacja

Graficzna reprezentacja obiektu (robot) zbudowana jest z prostych prymitywów geometrycznych:

- **Wózek:** obiekt typu `Rectangle`, którego pozycja pozioma aktualizowana jest w każdej klatce na podstawie zmiennej stanu $x(t)$.
- **Koła:** obiekty `Circle`, poruszające się wraz z wózkiem.
- **Wahadło:** obiekt liniowy, którego współrzędne końcowe wyznaczone są trygonometrycznie na podstawie kąta $\theta(t)$.

Kluczowym elementem implementacji jest funkcja aktualizująca `update`, wywoływana dla każdego kroku czasowego. Odpowiada ona za przeliczenie współrzędnych kinematycznych oraz przesunięcie okna widoku kamery tak, aby wózek znajdował się zawsze w centrum, co pozwala na obserwację ruchu na długim dystansie. Dodatkowo rysowany jest „ślad” (ang. *trail*) przebytej drogi przez oś wózka, co ułatwia wizualną ocenę stabilności pozycji.

Wykorzystanie animacji pozwala na szybką, intuicyjną weryfikację poprawności modelu fizycznego oraz ocenę jakości regulacji w sposób trudny do uchwycenia na statycznych wykresach (np. nienaturalne drgania czy gwałtowne reakcje „szarpnięcia”).



Rysunek 3.3. Zrzut ekranu z animacji realizowanej w środowisku Python (biblioteka Matplotlib). Widoczny wózek, wahadło oraz zakres ruchu.

4. Algorytmy sterowania

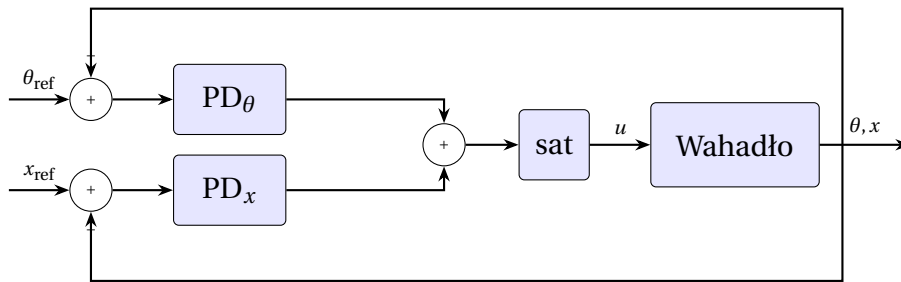
W niniejszym rozdziale przedstawiono szczegółowy opis algorytmów sterowania zaimplementowanych i przeanalizowanych w ramach pracy. Kod sterowników został zrealizowany w języku Python w postaci klas dziedziczących wspólną strukturę, co zapewnia modularność i łatwą wymiennność w pętli symulacyjnej. Każdy regulator wyznacza sygnał sterujący $u(t)$ (siłę przyłożoną do wózka) na podstawie aktualnego wektora stanu $x(t) = [\theta, \dot{\theta}, x, \dot{x}]^T$ oraz wartości zadanych x_{ref} .

W literaturze problem sterowania wahadłem odwróconym jest szeroko omawiany jako klasyczny problem testowy dla metod sterowania liniowego i nieliniowego [5], [7]. Poniżej opisano teoretyczne podstawy oraz szczegóły implementacyjne zbadanych struktur sterowania.

4.1. Równoległy regulator PID

Pierwszym zaimplementowanym układem jest regulator o strukturze kaskadowej lub równoległej, wykorzystujący klasyczne sprzężenie zwrotne typu PD (Proporcjonalno-Różniczkujące). W literaturze podejście to jest często stosowane jako punkt odniesienia dla bardziej zaawansowanych metod [4], [5].

W klasie `PDPDController` zastosowano strukturę równoległą, w której całkowity sygnał sterujący jest sumą reakcji na błąd kąta oraz błąd pozycji. Jest to podejście intuicyjne, dekomponujące problem na dwa podzadania: stabilizację wahadła w pozycji pionowej oraz doprowadzenie wózka do zadanej pozycji. Schemat blokowy regulatora przedstawiono na Rys. 4.1.



Rysunek 4.1. Schemat blokowy regulatora PID o strukturze równoległej.

Prawo sterowania wyraża się wzorem:

$$u(t) = \text{sat}_{u_{\max}}(u_\theta(t) + u_x(t)), \quad (24)$$

gdzie funkcja nasycenia $\text{sat}(\cdot)$ wynika z ograniczeń fizycznych siłownika. Definiując uchyby regulacji jako $e_\theta(t) = \theta_{\text{ref}} - \theta(t)$ oraz $e_x(t) = x_{\text{ref}} - x(t)$, prawo sterowania dla poszczegól-

nich pętli można zapisać w ogólnej postaci regulatora PID:

$$u_{\theta}(t) = K_{p,\theta} e_{\theta}(t) + K_{i,\theta} \int_0^t e_{\theta}(\tau) d\tau + K_{d,\theta} \frac{de_{\theta}(t)}{dt}, \quad (25)$$

$$u_x(t) = K_{p,x} e_x(t) + K_{i,x} \int_0^t e_x(\tau) d\tau + K_{d,x} \frac{de_x(t)}{dt}. \quad (26)$$

W powyższych równaniach przyjęto upraszczające założenie, że docelowe prędkości ($\dot{\theta}_{\text{ref}}, \dot{x}_{\text{ref}}$) wynoszą zero.

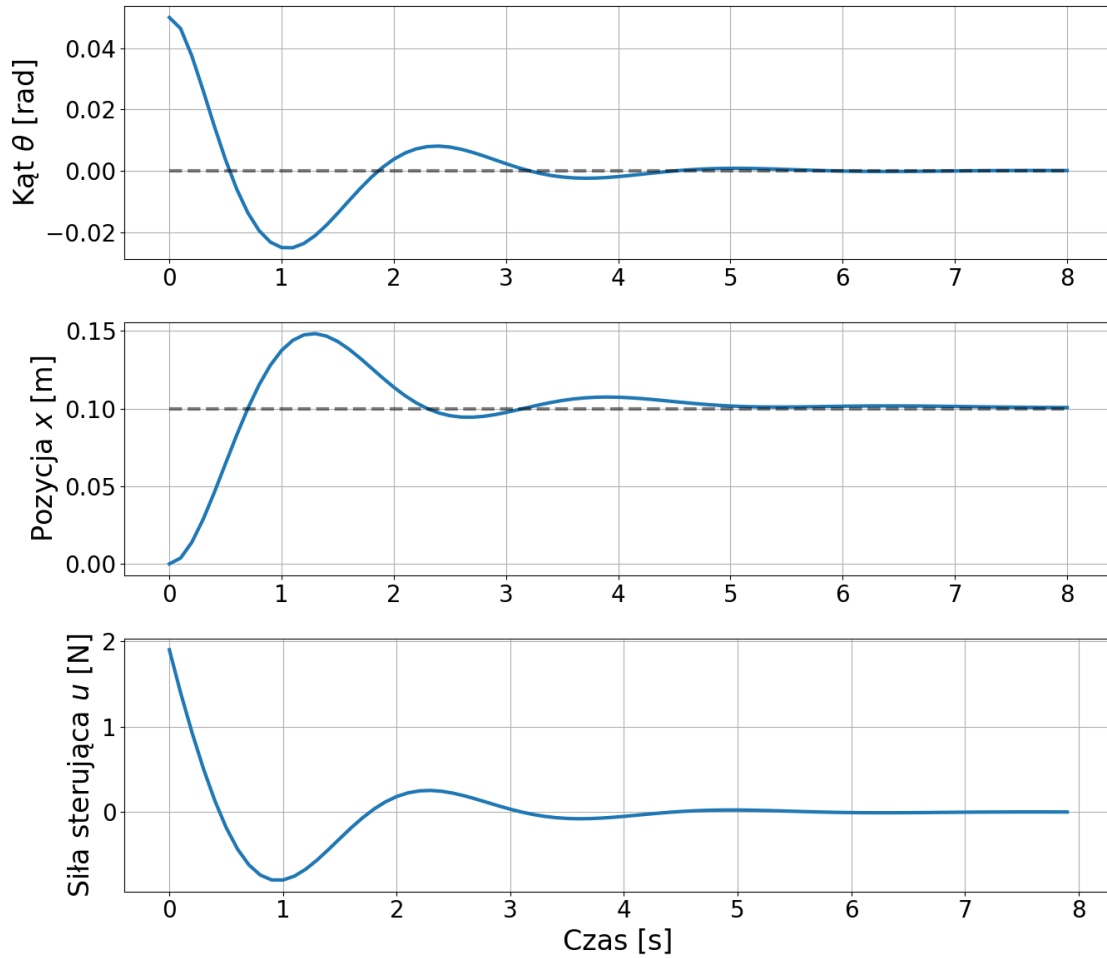
W implementacji programowej (plik `pd_pd.py`) przyjęto następujące nastawy dobrane eksperymentalnie:

- Tor stabilizacji kąta: $K_{p,\theta} = -95.0$, $K_{d,\theta} = -14.0$. Ujemne znaki wynikają z przyjętej konwencji układu współrzędnych i zwrotu siły.
- Tor pozycji: $K_{p,x} = -16.0$, $K_{d,x} = -14.0$.

Mimo iż klasa umożliwia włączenie członu całkującego (PID), w badaniach [6] często wskazuje się, że dla obiektów tej klasy człon różniczkujący (PD) jest kluczowy dla tłumienia oscylacji, a całkowanie może wprowadzać niestabilność w stanach nieustalonych bez odpowiednich mechanizmów anti-windup.

4.1.1. Proces doboru nastaw oraz analiza PID

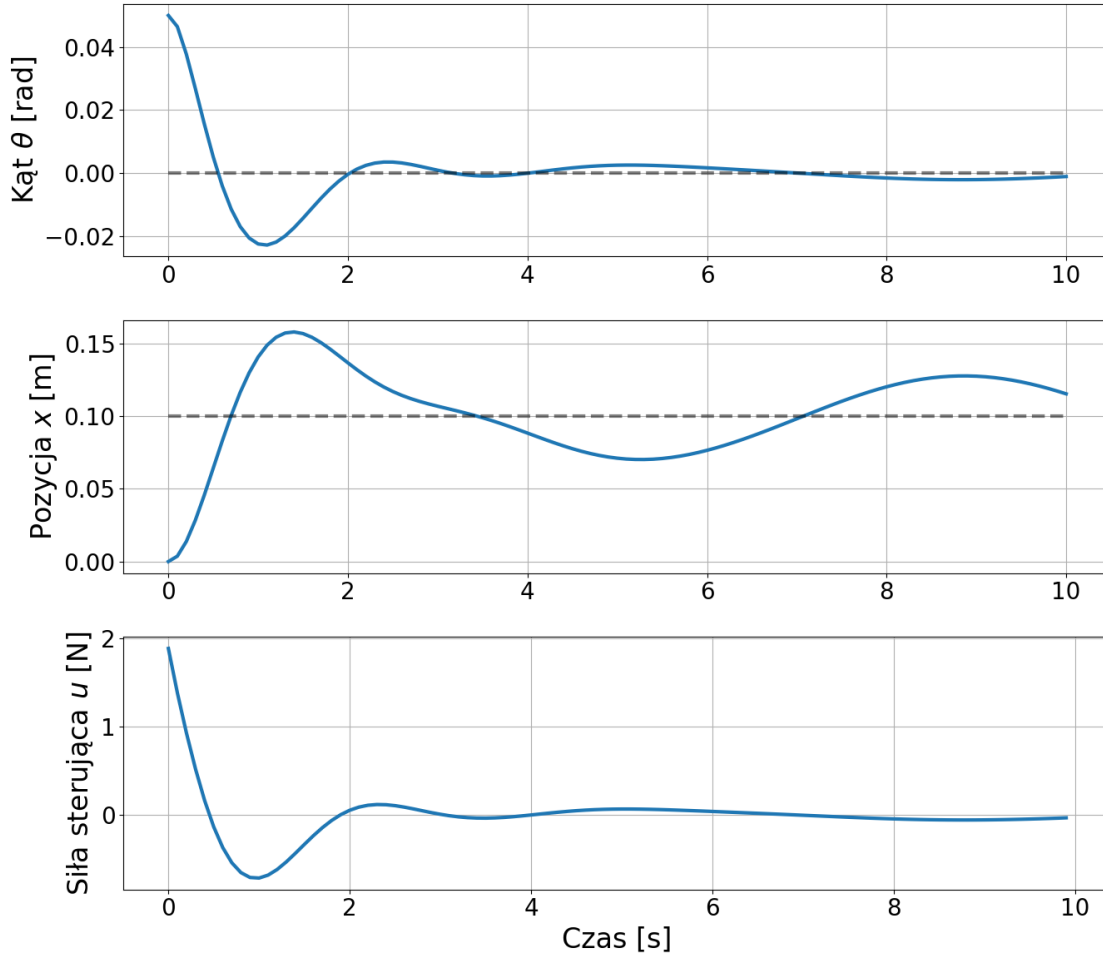
Dobór nastaw dla regulatora PID został zrealizowany wieloetapowo, ewoluując od metod heurystycznych do pełnej optymalizacji numerycznej.



Rysunek 4.2. Regulator PD nastawiony ręcznie ($K_{p,\theta} = -40$, $K_{d,\theta} = -8$, $K_{p,x} = -1$, $K_{d,x} = -3$).

Wstępne próby doboru metodą „prób i błędów” (Rys. 4.2), oparte na dekompozycji problemu (najpierw stabilizacja wahadła, potem pozycja wózka), pozwoliły uzyskać stabilność, jednak jakość regulacji była niezadowalająca. Układ charakteryzował się powolnym dochodzeniem do punktu pracy i znacznymi oscylacjami. Wzmocnienia proporcjonalne ($K_p = -40$ dla kąta) były niewystarczające, aby szybko tłumić odchylenia od pionu.

Analiza porównawcza struktur PD i PID W literaturze przedmiotu [6], [7] często podkreśla się, że dla obiektów niestabilnych statycznie, takich jak wahadło odwrócone, kluczowa jest szybka reakcja na zmiany kąta, którą zapewnia człon różniczkujący (D). Włączenie członu całkującego (I), tworzącego regulator PID, wprowadza dodatkowe przesunięcie fazowe (opóźnienie), co w układzie o dynamice astatycznej i nieliniowej prowadzi do znacznego pogorszenia zapasu stabilności.



Rysunek 4.3. Regulator PID z członem całkującym ($K_{p,\theta} = -40$, $K_{i,\theta} = -2$, $K_{d,\theta} = -8$, $K_{p,x} = -1$, $K_{i,x} = -2$, $K_{d,x} = -3$).

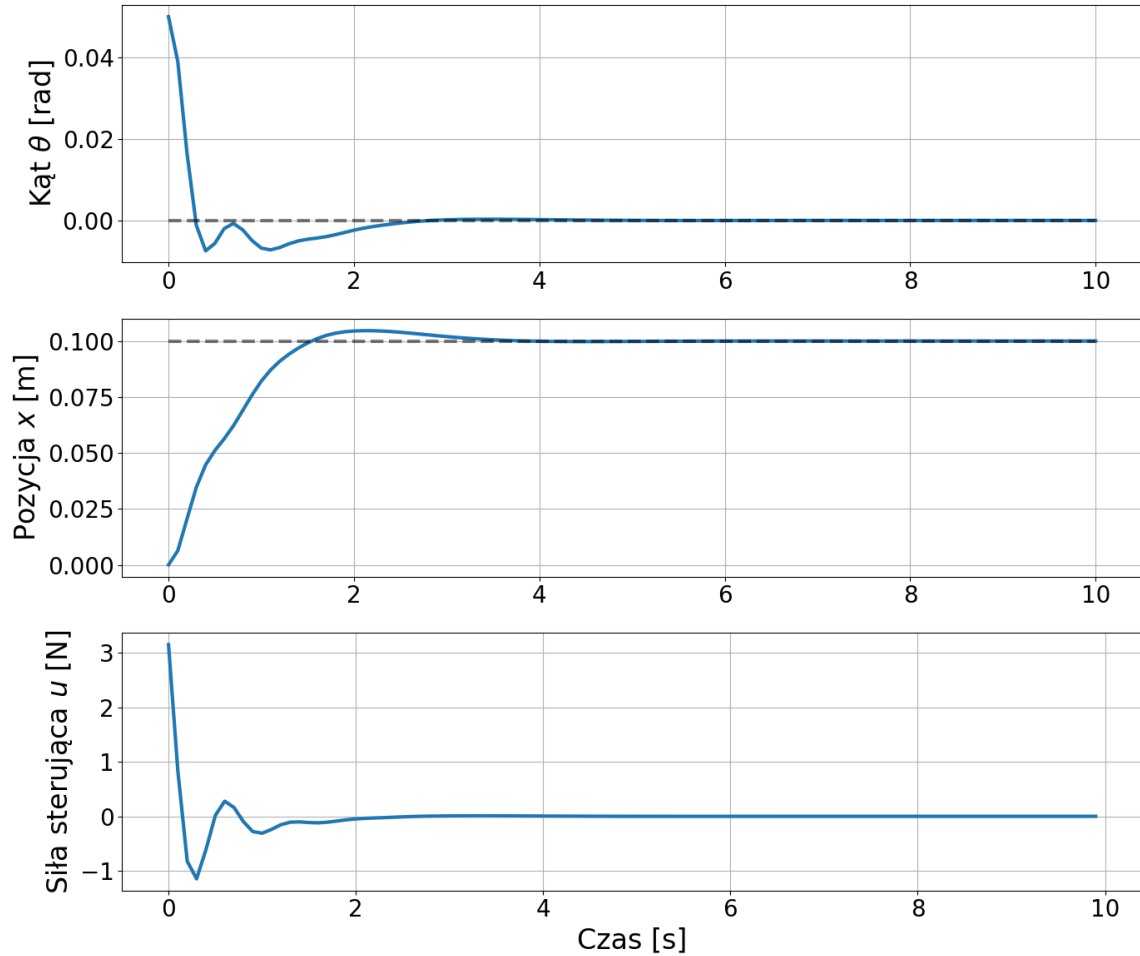
Przeprowadzone badania symulacyjne potwierdziły te wnioski (Rys. 4.3). Próba dodania akcji całkującej ($K_i \neq 0$) skutkowało zjawiskiem *integral wind-up* – akumulacją błędów w fazie rozruchu, co powodowało przeregulowania wykraczające poza obszar przyciągania stabilnego punktu równowagi. Regulator PID wpadał w niegasnące oscylacje lub doprowadzał do przewrócenia wahadła, podczas gdy „czysty” regulator PD zapewnia sztywne i szybkie sterowanie.

Aby wyeliminować subiektywność strojenia ręcznego, zastosowano algorytm Ewolucji Różnicowej (Differential Evolution), zaimplementowany w module `scipy.optimize`. Zdefiniowano globalną funkcję kosztu J , która gwarantuje „uczciwe” porównanie wszystkich badanych regulatorów:

$$J = w_\theta \cdot \text{MSE}(\theta) + w_x \cdot \text{MSE}(x) + w_u \cdot \text{RMS}(u), \quad (27)$$

gdzie przyjęto wagi $w_\theta = 4.0$ (priorytet stabilizacji), $w_x = 1.0$ (dokładność pozycjonowania)

oraz $w_u = 0.01$ (koszt energii). Algorytm operował na populacji 10 osobników przez 20 generacji, co pozwoliło uniknąć minimów lokalnych.

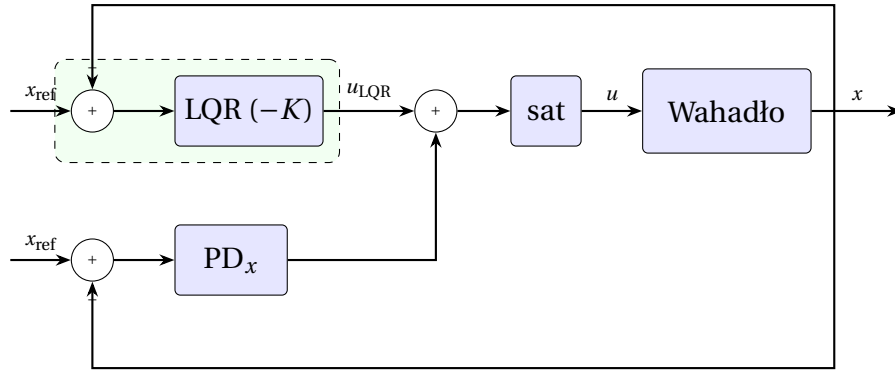


Rysunek 4.4. Zoptymalizowany regulator PID ($K_{p,\theta} = -95$, $K_{d,\theta} = -14$, $K_{p,x} = -16$, $K_{d,x} = -14$).

Zoptymalizowane nastawy (Rys. 4.4) charakteryzują się znacznie wyższymi wzmocnieniami niż dobrane ręcznie: $K_{p,\theta} = -95$ (vs. -40) oraz $K_{p,x} = -16$ (vs. -1). Algorytm „odkrył”, że agresywna reakcja na błąd pozycji wózka pośrednio stabilizuje wahadło, ponieważ wymusza szybkie korekty trajektorii.

4.2. Układ hybrydowy PID-LQR

Kolejnym analizowanym algorytmem jest regulator liniowo-kwadratowy (LQR), będący standardem w sterowaniu optymalnym systemów wielowymiarowych MIMO [8]. Klasa `PDLQRController` implementuje sterowanie oparte na pełnym wektorze stanu, wspomagane dodatkowym członem PD dla uchybu pozycji, co tworzy strukturę hybrydową opisaną m.in. w [5] oraz [7] (w kontekście porównawczym). Schemat blokowy tego układu przedstawiono na Rys. 4.5.



Rysunek 4.5. Schemat blokowy hybrydowego regulatora PID-LQR.

Problem LQR polega na znalezieniu prawa sterowania $u(t) = -Kx(t)$, które minimalizuje wskaźnik jakości:

$$J = \int_0^{\infty} (x(t)^T Q x(t) + u(t)^T R u(t)) dt, \quad (28)$$

gdzie $Q \geq 0$ jest macierzą wag stanu, a $R > 0$ wagą sterowania. Optymalna macierz wzmocnień K wyznaczana jest poprzez rozwiązanie algebraicznego równania Riccatiego (CARE):

$$A^T P + P A - P B R^{-1} B^T P + Q = 0, \quad (29)$$

skąd $K = R^{-1} B^T P$. Macierze A i B pochodzą z linearyzacji modelu wahadła wokół punktu równowagi górnej ($\theta = 0$).

W zaimplementowanym rozwiązaniu (plik `pd_lqr.py`), sygnał sterujący składa się z dwóch komponentów:

$$u(t) = u_{\text{LQR}}(t) + u_{\text{PD,pos}}(t). \quad (30)$$

Składnik LQR realizuje stabilizację wokół punktu pracy:

$$u_{\text{LQR}}(t) = -K \cdot (x(t) - x_{\text{ref}}). \quad (31)$$

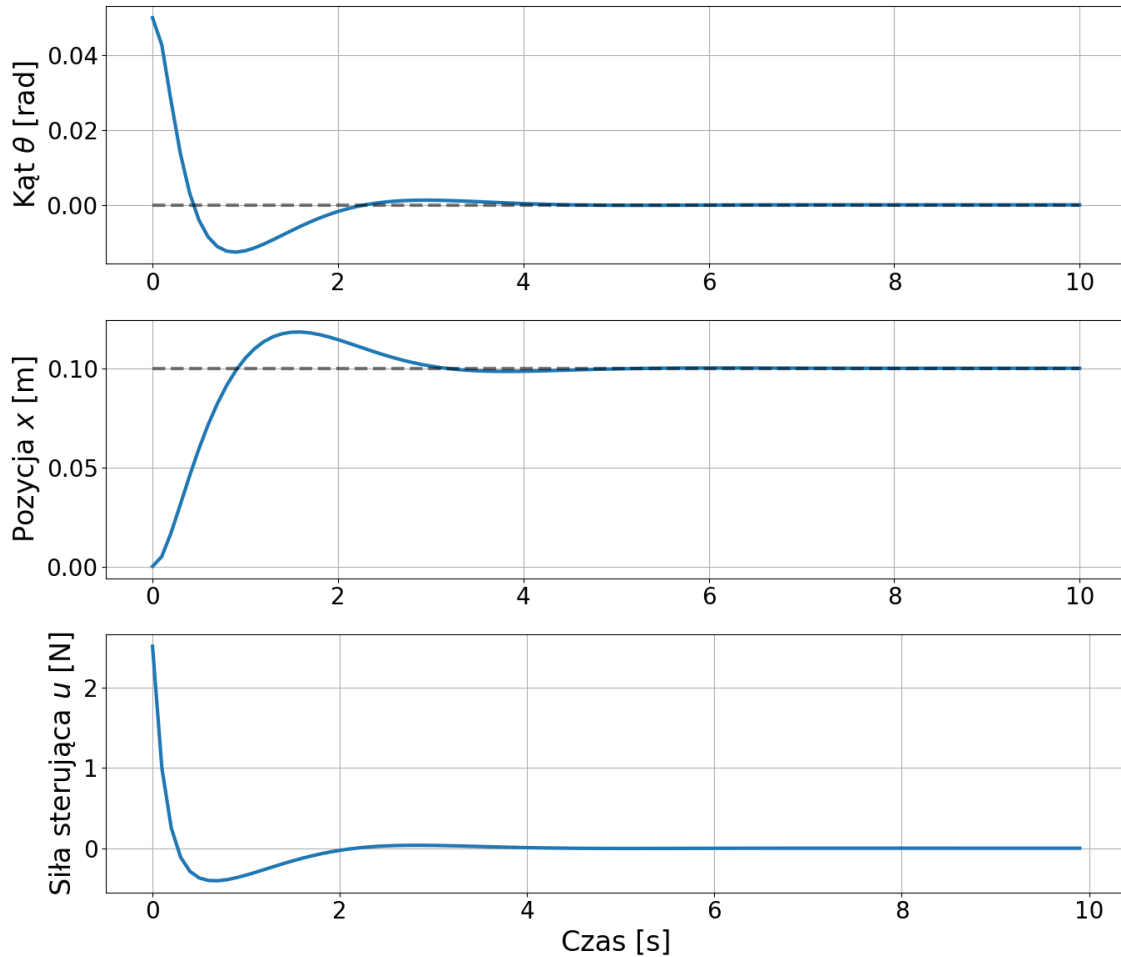
Zastosowane wagi optymalne to:

$$Q = \text{diag}([1.0, 1.0, 500.0, 250.0]), \quad R = 1.0. \quad (32)$$

Dodatkowy człon PD na pętli pozycji (zrealizowany analogicznie do wzoru 26) ma na celu poprawę śledzenia skokowych zmian wartości zadanej x_{ref} , co jest częstą praktyką w aplikacjach praktycznych, gdzie LQR zapewnia stabilność, a regulator zewnętrzny dba o uchyb w stanie ustalonym [6].

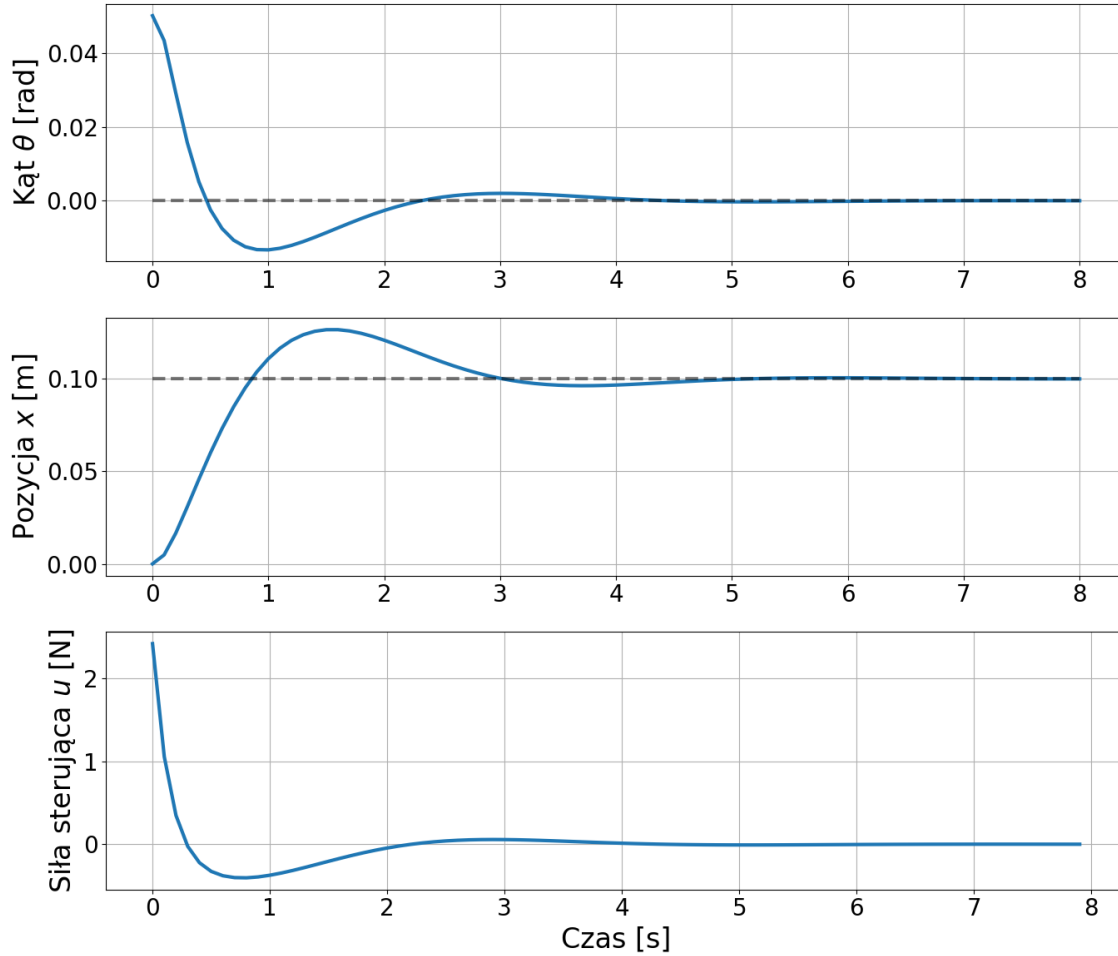
4.2.1. Dobór wag macierzy Q i R

Dobór wag dla regulatora LQR również charakteryzował się ewolucyjnym podejściem do problemu optymalizacji wskaźnika jakości.



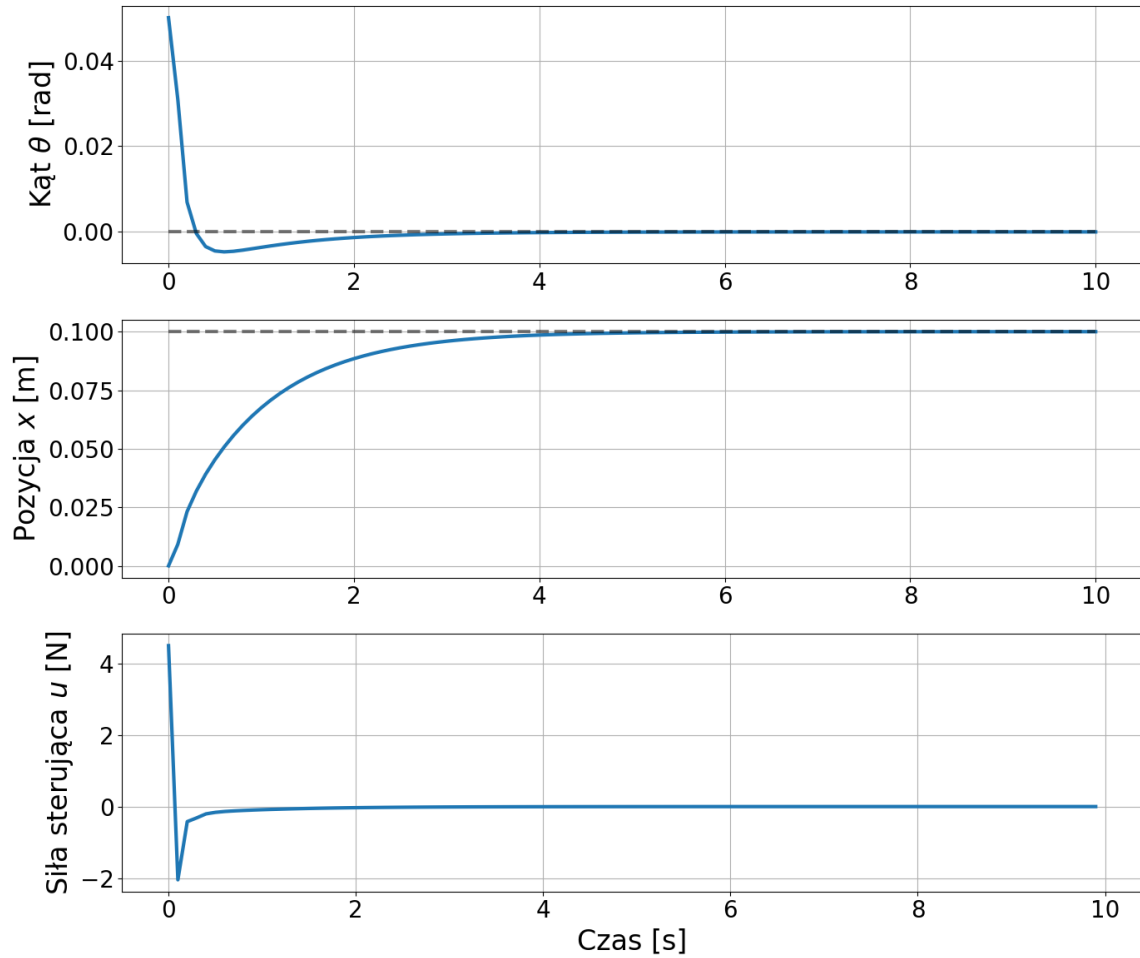
Rysunek 4.6. Regulator LQR z wagami jednostkowymi ($Q = I$, $R = 1$, PD: $K_{p,x} = -4.5$, $K_{d,x} = -3$).

W pierwszej fazie przyjęcie jednostkowej macierzy diagonalnej $Q = I$ oraz $R = 1$ (Rys. 4.6) okazało się niewystarczające. Mimo teoretycznej stabilności wynikającej z rozwiązania równania CARE, wahadło wykonywało bardzo duże wychylenia, a wózek wielokrotnie wyjeżdżał poza dopuszczalny zakres roboczy toru. Problem wynikał z faktu, że jednostkowe wagi traktują 1 rad błędu kąta tak samo jak 1 m błędu pozycji i 1 N^2 kosztu sterowania – co jest fizycznie nieuzasadnione.



Rysunek 4.7. Regulator LQR strojony metodą Brysona ($Q = \text{diag}([25, 1, 4, 1])$, $R = 10$, PD: $K_{p,x} = -4.5$, $K_{d,x} = -3$).

Następnie przeprowadzono strojenie ręczne metodą prób i błędów, inspirowane regułą Brysona (Rys. 4.7). Reguła ta postuluje, że elementy diagonalne macierzy Q i R powinny być odwrotnie proporcjonalne do kwadratów maksymalnych dopuszczalnych wartości odpowiednich zmiennych stanu i sterowania, tj. $Q_{ii} = 1/x_{i,\max}^2$ oraz $R = 1/u_{\max}^2$. Ręczne zwiększanie kar za wychylenie kąta ($Q_\theta = 25$) poprawiło sztywność wahadła. Udało się ustalić zestaw wag zapewniający stabilną pracę, choć czas regulacji był wciąż niezadowalający, a reakcja na zakłócenia powolna.

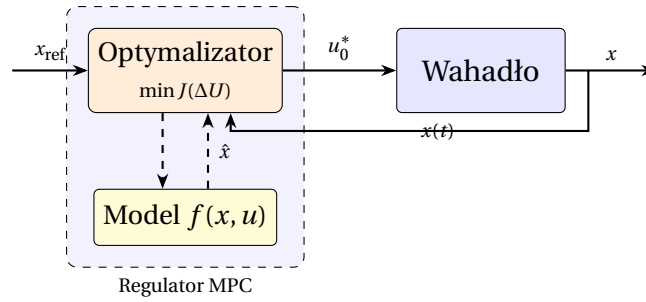


Rysunek 4.8. Zoptymalizowany regulator PID-LQR ($Q = \text{diag}([1, 1, 500, 250])$, $R = 1$, PID: $K_{p,x} = -1.5$, $K_{d,x} = -5$).

W ostatnim etapie zastosowano optymalizację numeryczną (Rys. 4.8). Algorytm genetyczny poszukiwał optymalnych elementów diagonalnych macierzy Q oraz skalaru R , minimalizując wskaźnik jakości. Zoptymalizowane wagi (w szczególności wysoka kara $Q_x = 500$) sprawiają, że regulator bardzo agresywnie pilnuje pozycji wózka, co pośrednio wymusza stabilne trzymanie wahadła. Interesujące jest, że optymalizator „odkrył” strategię przeciwną do intuicyjnej: zamiast karać głównie kąt (Q_θ), kładzie nacisk na pozycję (Q_x), co zmusza wózek do szybkich korekt stabilizujących wahadło.

4.3. Nieliniowe sterowanie predykcyjne (MPC)

Algorytm MPC (Model Predictive Control) stanowi zaawansowaną metodę sterowania, która w odróżnieniu od LQR, uwzględnia wprost ograniczenia sygnału sterującego oraz nieliniową dynamikę obiektu [9], [14]. Zaimplementowany w klasie `MPCController` (plik `mpc.py`) algorytm rozwiązuje w każdym kroku symulacji problem optymalizacji dynamicznej nieliniowej (NMPC). Zasadę działania regulatora MPC ilustruje Rys. 4.9.



Rysunek 4.9. Schemat blokowy regulatora MPC z wewnętrznym modelem predykcyjnym.

Zadanie optymalizacji rozwiązywane jest numerycznie metodą SQP (Sequential Quadratic Programming) przy użyciu solwera SLSQP z biblioteki `scipy.optimize`. Wybór tego solwera podyktowany był jego dostępnością w standardowej dystrybucji Pythona oraz zdolnością do obsługi ograniczeń nierównościowych (wymaganych dla saturacji sterowania). Zadanie zdefiniowane jest następująco:

$$\min_{\Delta U} J = \sum_{k=1}^{N_p} (\hat{x}_k - x_{\text{ref}})^T Q (\hat{x}_k - x_{\text{ref}}) + R \sum_{k=0}^{N_c-1} (\Delta u_k)^2, \quad (33)$$

przy ograniczeniach:

$$\hat{x}_{k+1} = f(\hat{x}_k, u_k), \quad k = 0, \dots, N_p - 1 \quad (34)$$

$$u_{\min} \leq u_k \leq u_{\max}, \quad (35)$$

$$u_k = u_{k-1} + \Delta u_k. \quad (36)$$

Gdzie:

- $N_p = 12$ – horyzont predykcji,
- $N_c = 4$ – horyzont sterowania. Dla kroków $k \geq N_c$ stosowane jest tzw. *blokowanie sterowania*, co oznacza, że przyrosty sterowania $\Delta u_k = 0$ i sygnał sterujący pozostaje stały: $u_k = u_{N_c-1}$. Technika ta redukuje liczbę zmiennych decyzyjnych z N_p do N_c , przyspieszając obliczenia przy zachowaniu długiego horyzontu predykcji,
- $f(\cdot)$ – nieliniowy model dyskretny obiektu (całkowanie metodą Rungego-Kutta 4. rzędu),
- $Q = \text{diag}([158.4, 36.8, 43.4, 19.7])$ – macierz kar stanu,
- $R = 0.086$ – współczynnik kary za zmianę sterowania (Δu).

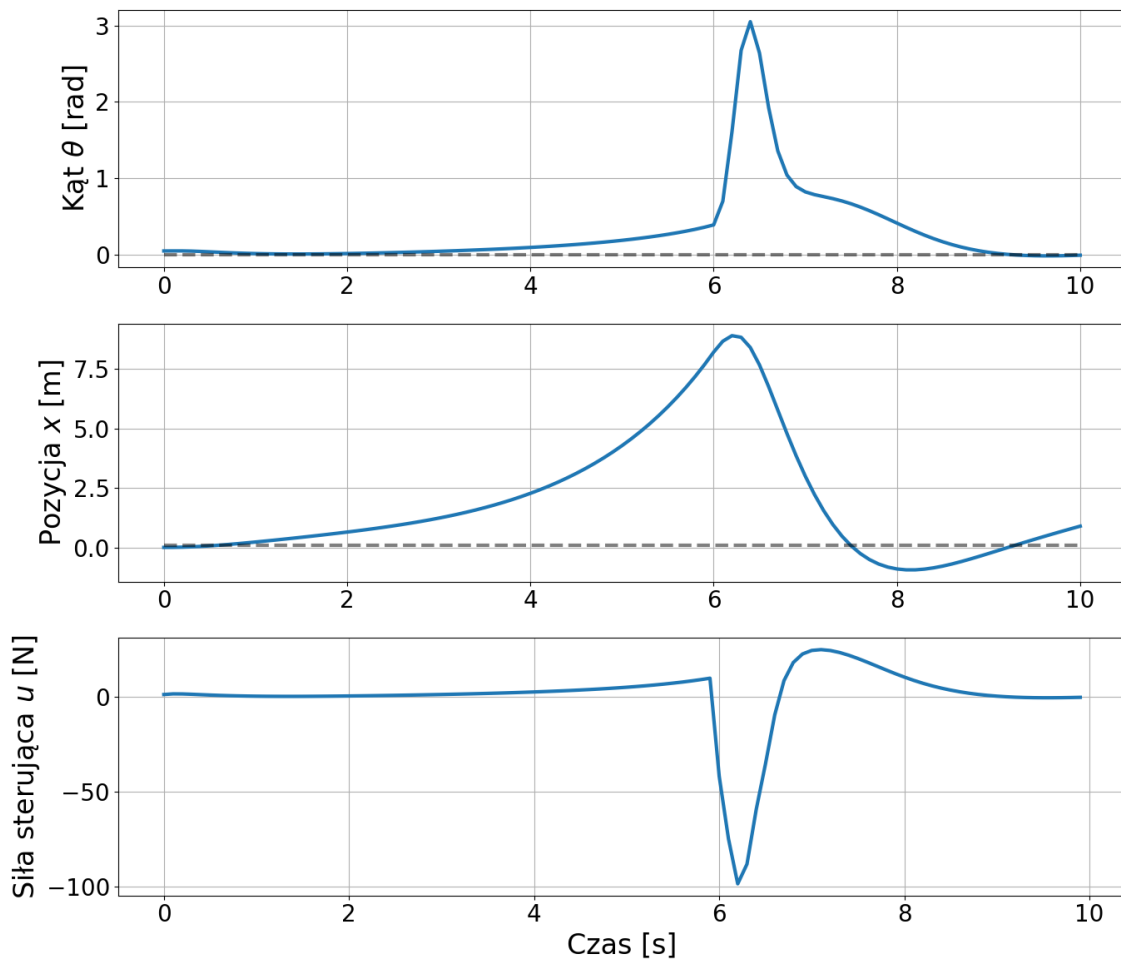
Kluczową zaletę MPC, podkreślaną w pracach [11] oraz [8], jest możliwość bezpośredniego uwzględnienia ograniczeń (saturacji) już na etapie wyliczania sterowania, co zapobiega zjawisku nasycenia elementu wykonawczego, które mogłoby mieć miejsce w przypadku LQR.

Analiza wykazała, że bezpośrednie przeniesienie macierzy wag Q i R z regulatora LQR do sterownika MPC prowadziło do znaczącego pogorszenia jakości sterowania (wydłuże-

nie czasu regulacji z ok. 3s do ponad 9s). Wynika to z faktu, że model MPC, dzięki jawnemu uwzględnieniu ograniczeń sygnału sterującego, pozwala na zastosowanie znacznie bardziej agresywnych nastaw (większych kar za błędy stanu), które w liniowym regulatorze LQR powodowałyby nasycenie i potencjalną niestabilność. Dlatego zdecydowano się na niezależną optymalizację parametrów obu regulatorów, aby porównywać ich najlepsze możliwe konfiguracje, a nie identyczne, ale nieoptymalne nastawy.

4.3.1. Dobór horyzontu i wag funkcji celu

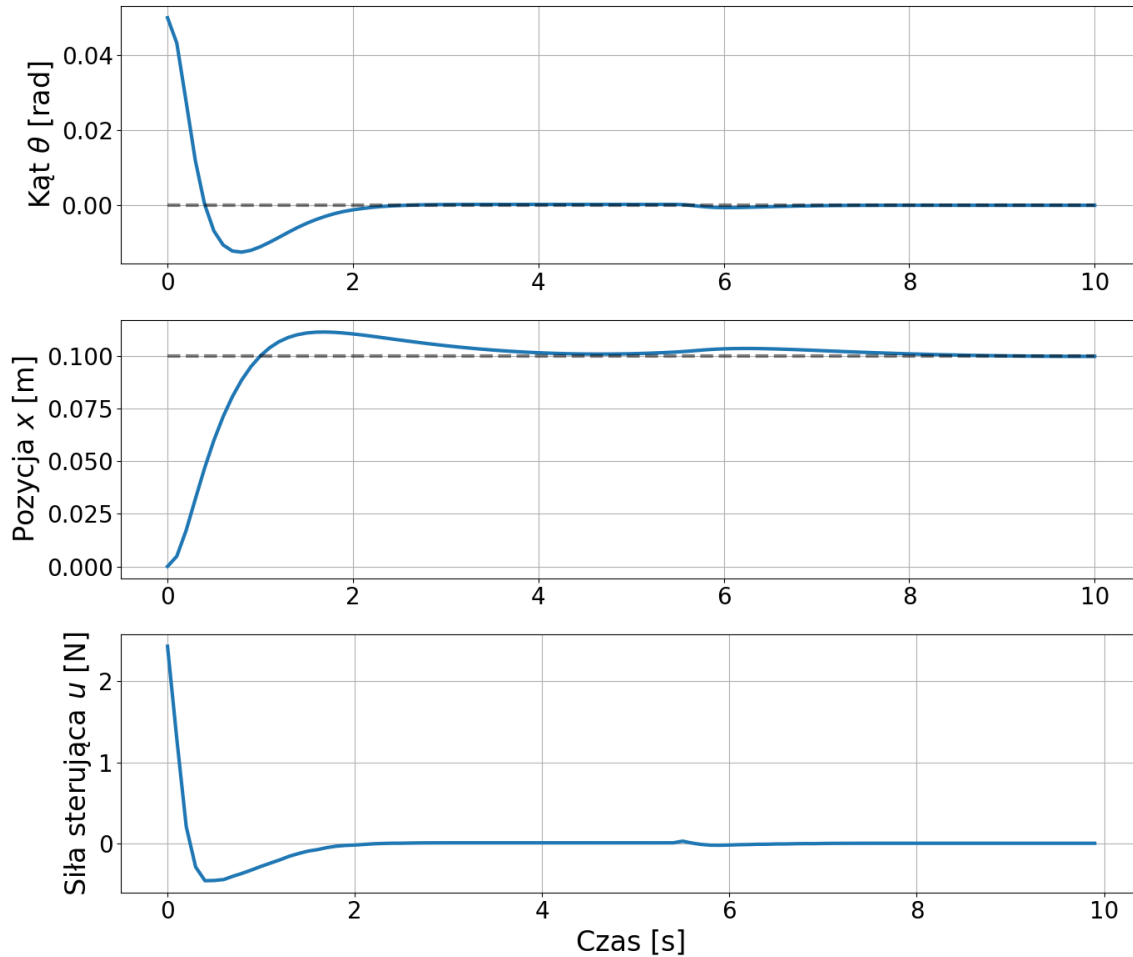
Dla regulatora MPC kluczowym wyzwaniem był dobór horyzontu predykcji oraz macierzy wag, determinujących zachowanie układu w stanie nieustalonym.



Rysunek 4.10. Regulator MPC z krótkim horyzontem ($N_p = 5$, $N_c = 2$, $Q = \text{diag}([10, 1, 10, 1])$, $R = 0.1$).

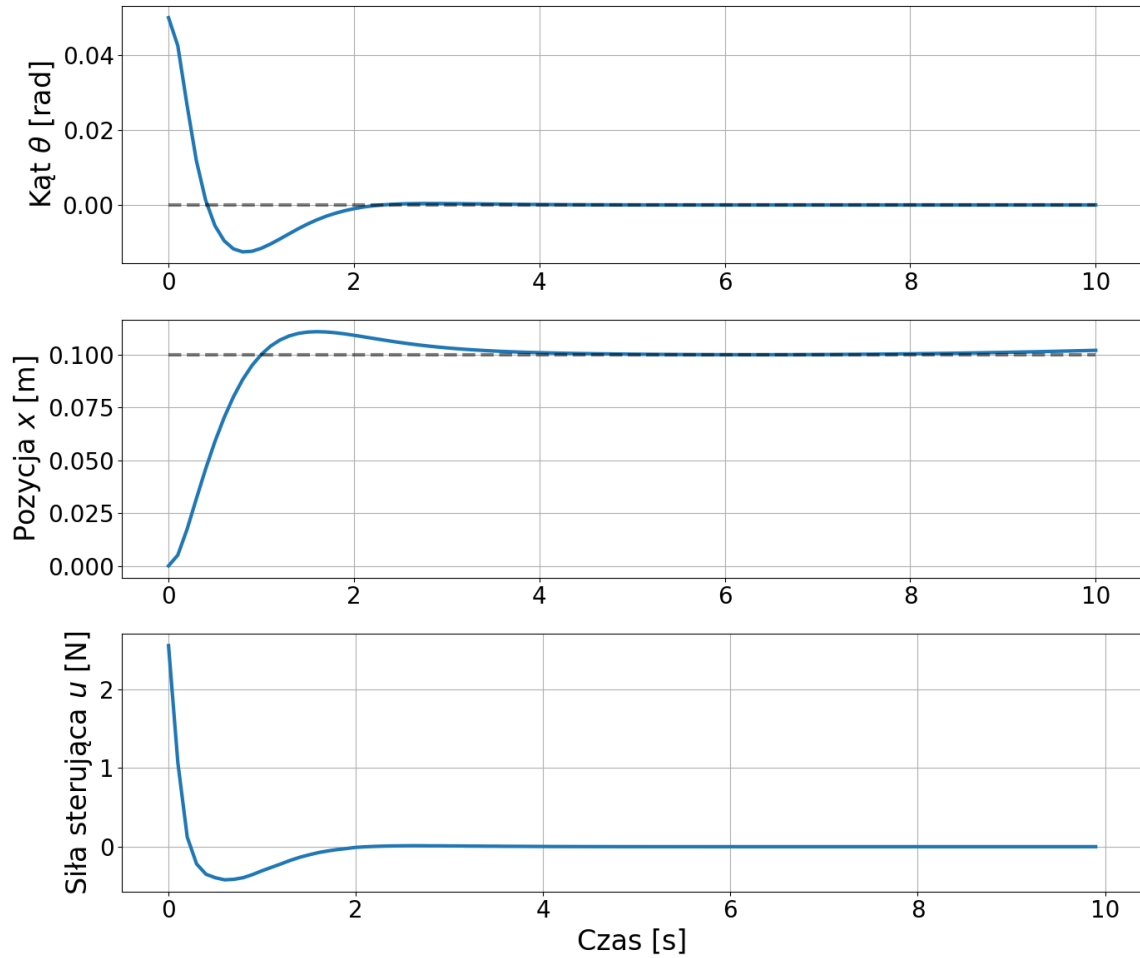
Początkowe ustawienie zbyt krótkiego horyzontu predykcji ($N_p = 5$, Rys. 4.10) prowadziło do niestabilności układu zamkniętego. Regulator „nie widział”, że rozpędzając wózek w celu korekcji kąta, nie zdąży wyhamować przed upadkiem wahadła lub osiągnięciem końca toru. Horyzont pięciu kroków (przy $\Delta t = 0,01$ s daje zaledwie 0,05 s predykcji)

jest niewystarczający, aby uchwycić pełną dynamikę wahadła i zaplanować odpowiedni manewr powrotny.



Rysunek 4.11. Regulator MPC z ręcznie dobranymi wagami ($N_p = 10$, $N_c = 3$, $Q = \text{diag}([50, 10, 50, 10])$, $R = 0.1$).

Zwiększenie horyzontu do $N_p = 10$ w ramach korekty ręcznej (Rys. 4.11) ustabilizowało proces. Regulator zyskał możliwość „patrzenia w przyszłość” na 0,1 s, co pozwoliło na antycypację skutków swoich działań. Dodatkowa manipulacja wagami Q pozwoliła na uzyskanie poprawnego sterowania, jednak odpowiedź dynamiczna była powolna, a przebiegi wykazywały przeregulowania.



Rysunek 4.12. Zoptymalizowany regulator MPC ($N_p = 12$, $N_c = 4$, $Q = \text{diag}([158.4, 36.8, 43.4, 19.7])$, $R = 0.086$).

Automatyzacja procesu strojenia przy użyciu skryptu `tune_mpc.py` pozwoliła na znalezienie kompromisu między długością horyzontu a wagami (Rys. 4.12). Algorytm optymalizacyjny wskazał $N_p = 12$ jako optimum dla tego modelu dyskretnego, zapewniając stabilność przy akceptowalnym czasie obliczeń. Zoptymalizowane wagi Q znacząco różnią się od intuicyjnych proporcji – wysoka kara za kąt ($Q_\theta = 158.4$) w połączeniu z niską karą za zmianę sterowania ($R = 0.086$) zapewnia agresywną, ale gładką stabilizację.

4.4. MPC z rozszerzonym wskaźnikiem jakości (MPC-J2)

Zaimplementowano sterownik `MPCControllerJ2` jako wariant badawczy algorytmu predykcyjnego (plik `mpc_J2.py`). Jego struktura jest zbliżona do podstawowego MPC, jednak funkcja kosztu została rozbudowana o dodatkowy składnik karzący bezwzględną wartość sygnału sterującego (energii), a nie tylko jego przyrosty.

Zmodyfikowana funkcja celu przyjmuje postać:

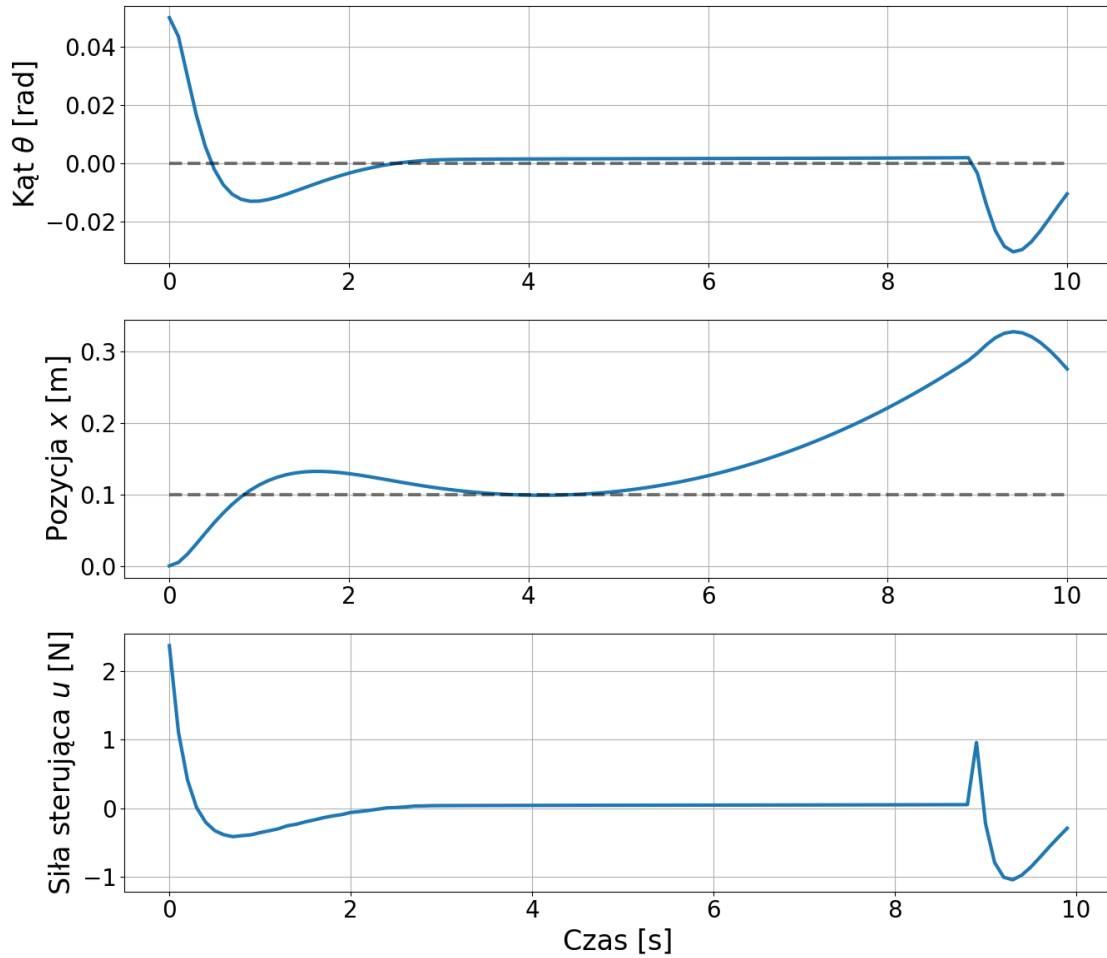
$$J = \sum_{k=1}^{N_p} (x_k - x_{\text{ref}})^T Q (x_k - x_{\text{ref}}) + R_{\Delta} \sum_{k=0}^{N_c-1} (\Delta u_k)^2 + R_{\text{abs}} \sum_{k=0}^{N_c-1} (u_k)^2. \quad (37)$$

Wprowadzenie parametru R_{abs} pozwala na bezpośrednie minimalizowanie zużycia energii sterowania, co jest podejściem powszechnie stosowanym w praktycznych implementacjach algorytmów predykcyjnych [9], [14]. Ograniczenie amplitudy sygnału sterującego nie tylko redukuje wydatek energetyczny (istotny w aplikacjach mobilnych), ale także zmniejsza obciążenie mechaniczne elementów wykonawczych, co wpływa na żywotność napędu.

W badaniach przyjęto zoptymalizowane wagi: $q_{\theta} = 40.0$, $q_x = 40.0$, $q_{\dot{\theta}} = 5.0$, $q_{\dot{x}} = 5.0$ (elementy macierzy diagonalnej Q), przy czym $R_{\Delta} = 0.0001$ i $R_{\text{abs}} = 0$ dla wariantu priorytetyzującego jakość regulacji.

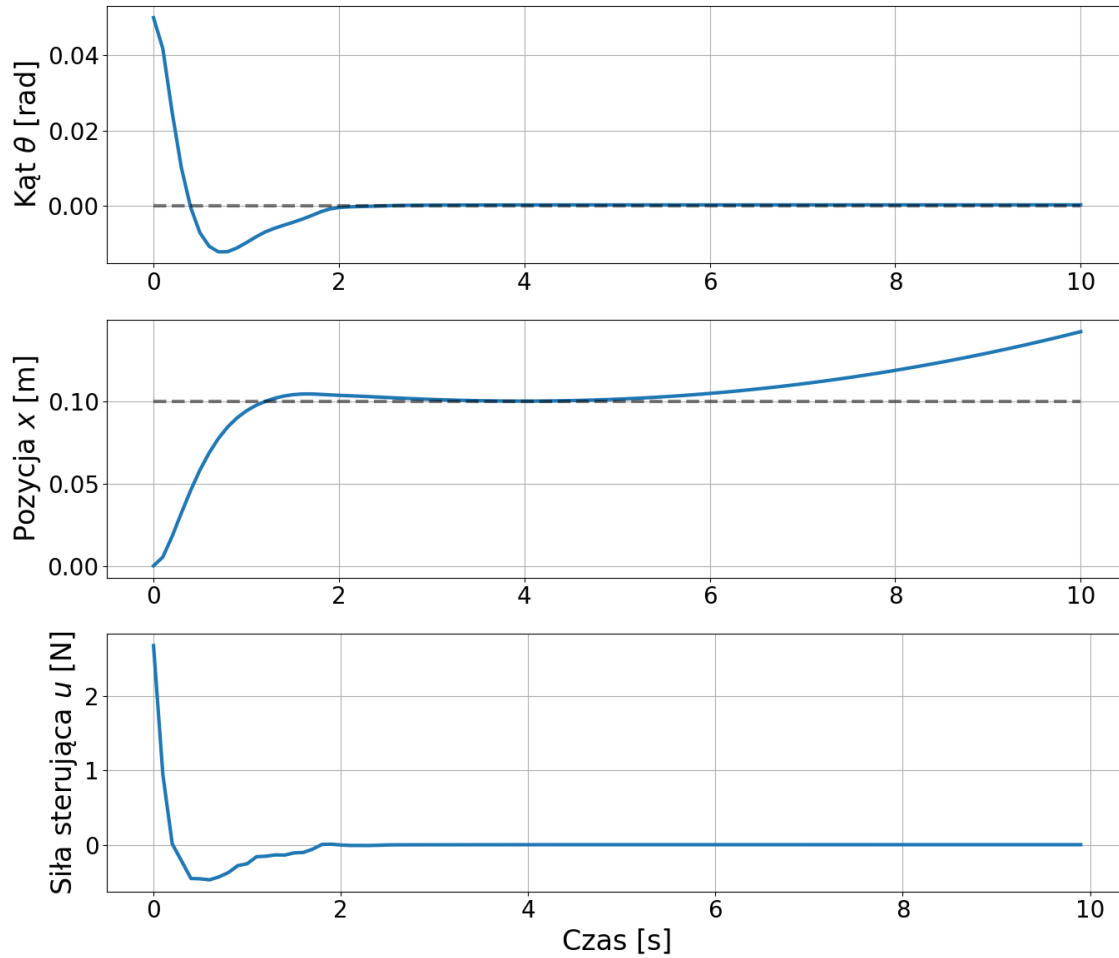
4.4.1. Analiza wpływu kary za energię

W przypadku wariantu MPC-J2 analizowano nieliniowy wpływ parametru R_{abs} na zachowanie układu. Eksperymenty przeprowadzono przy stałych wagach stanu ($q_{\theta} = 40$, $q_x = 40$, $q_{\dot{\theta}} = 5$, $q_{\dot{x}} = 5$), zmieniając jedynie wartość kary za bezwzględną wartość sterowania.



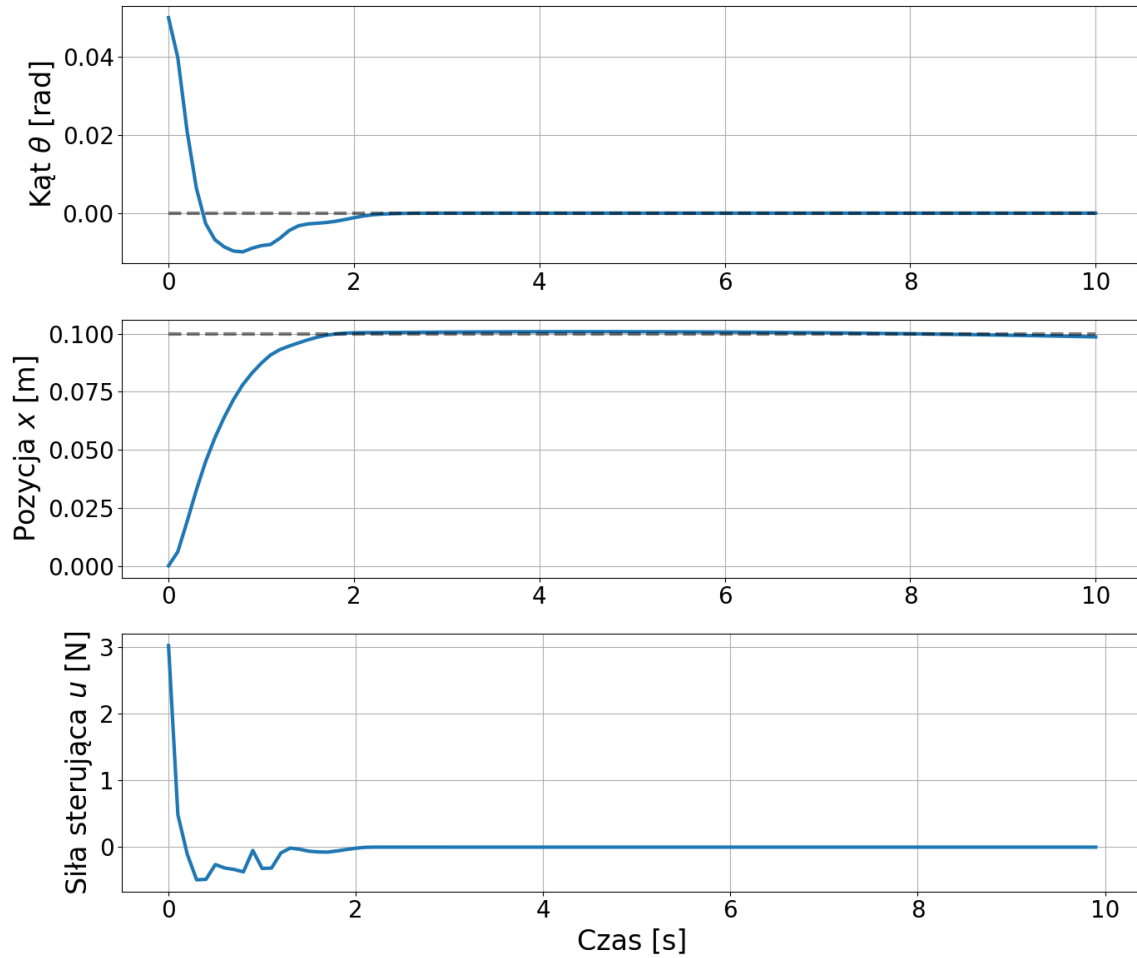
Rysunek 4.13. Regulator MPC-J2 z wysoką karą za energię ($R_{\text{abs}} = 10$).

Przyjęcie zbyt dużej wartości kary za sterowanie bezwzględne ($R_{\text{abs}} = 10$, Rys. 4.13) spowodowało, że regulator wykazywał tendencję do pasywności. Funkcja kosztu karała każdy niuton siły tak mocno, że optymalizator „wolał” pozwolić wahadłu upaść, niż ponieść wysoki koszt energetyczny ratowania go. W efekcie układ nie był w stanie ustabilizować się – wahadło przewracało się, ponieważ koszt energetyczny utrzymania go w pionie przewyższał zysk wynikający z małego błędu kąтового w funkcji celu.



Rysunek 4.14. Regulator MPC-J2 z ręcznie zmniejszoną karą ($R_{\text{abs}} = 1$).

Stopniowe, ręczne zmniejszanie parametru R_{abs} do wartości 1.0 (Rys. 4.14) pozwoliło na początkową stabilizację, jednak układ ostatecznie ulegał destabilizacji. Regulator działał zbyt zachowawczo – oszczędzając energię, pozwalał wózkowi na zbyt duży dryf od pozycji zadanej. Gdy błąd pozycji narastał, próba korekcji wymagała agresywnego ruchu wózka, co z kolei destabilizowało wahadło. Taki scenariusz ilustruje typowy problem źle dostrojonego kompromisu między jakością regulacji a oszczędnością energii.

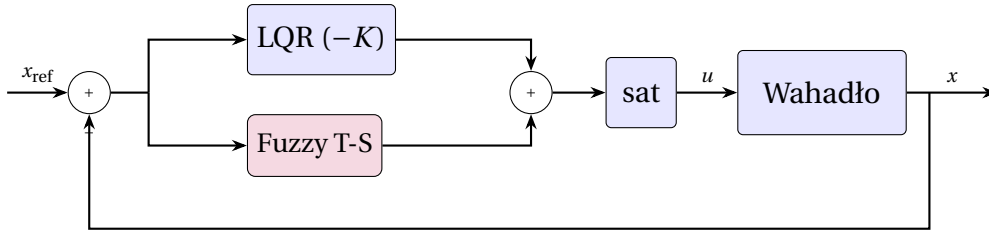


Rysunek 4.15. Zoptymalizowany regulator MPC-J2 ($R_{\text{abs}} = 0$).

Algorytm optymalizacyjny wskazał, że dla tego zadania najlepszym rozwiązaniem jest całkowite wyłączenie kary za bezwzględną wartość sterowania ($R_{\text{abs}} = 0$, Rys. 4.15). Pozostała jedynie kara za przyrosty sterowania $R_{\Delta} = 0.0001$, która zapewnia gładkość sygnału sterującego bez ograniczania zdolności regulatora do szybkiej reakcji. Znalezione „złoty środek”, w którym układ stabilizuje się szybko, a sterowanie pozbawione jest zbędnych oscylacji wysokoczęstotliwościowych.

4.5. Regulator rozmyty wspomagany LQR (Fuzzy-LQR)

Ostatnim zbadanym układem jest sterownik hybrydowy TSFuzzyController (plik `fuzzy_lqr.py`), łączący liniowy regulator LQR z systemem wnioskowania rozmytego typu Takagi-Sugeno (T-S). Koncepcja ta, opisana szerzej w [7] oraz [13], ma na celu adaptację wzmocnień regulatora w zależności od punktu pracy, co pozwala na agresywniejszą reakcję w przypadku dużych odchyłeń od pionu. Strukturę regulatora przedstawiono na Rys. 4.16.



Rysunek 4.16. Schemat blokowy regulatora Fuzzy-LQR z równoległą strukturą hybrydową.

Sygnał sterujący jest sumą:

$$u(t) = u_{\text{LQR}}(t) + u_{\text{Fuzzy}}(t). \quad (38)$$

Część rozmyta $u_{\text{Fuzzy}}(t)$ wykorzystuje bazę reguł postaci:

JEŚLI e_θ jest A_i ORAZ $\dot{\theta}$ jest B_i ... TO $u_i = f_i(x)$,

gdzie $f_i(x)$ jest liniową funkcją stanu (lokalny regulator liniowy). Zastosowano funkcje przynależności trójkątne dla zmiennych stanu, dzieląc przestrzeń na obszary „Mały błąd” i „Duży błąd”. Baza wiedzy składa się z 16 reguł (2^4 kombinacji dla 4 zmiennych stanu). Wyjście sterownika obliczane jest jako średnia ważona:

$$u_{\text{Fuzzy}} = G \cdot \frac{\sum_{i=1}^{16} w_i(x) \cdot u_i}{\sum_{i=1}^{16} w_i(x)}, \quad (39)$$

gdzie w_i to stopień aktywacji i -tej reguły, a $G = 0.37$ to globalne wzmocnienie skalujące (po optymalizacji).

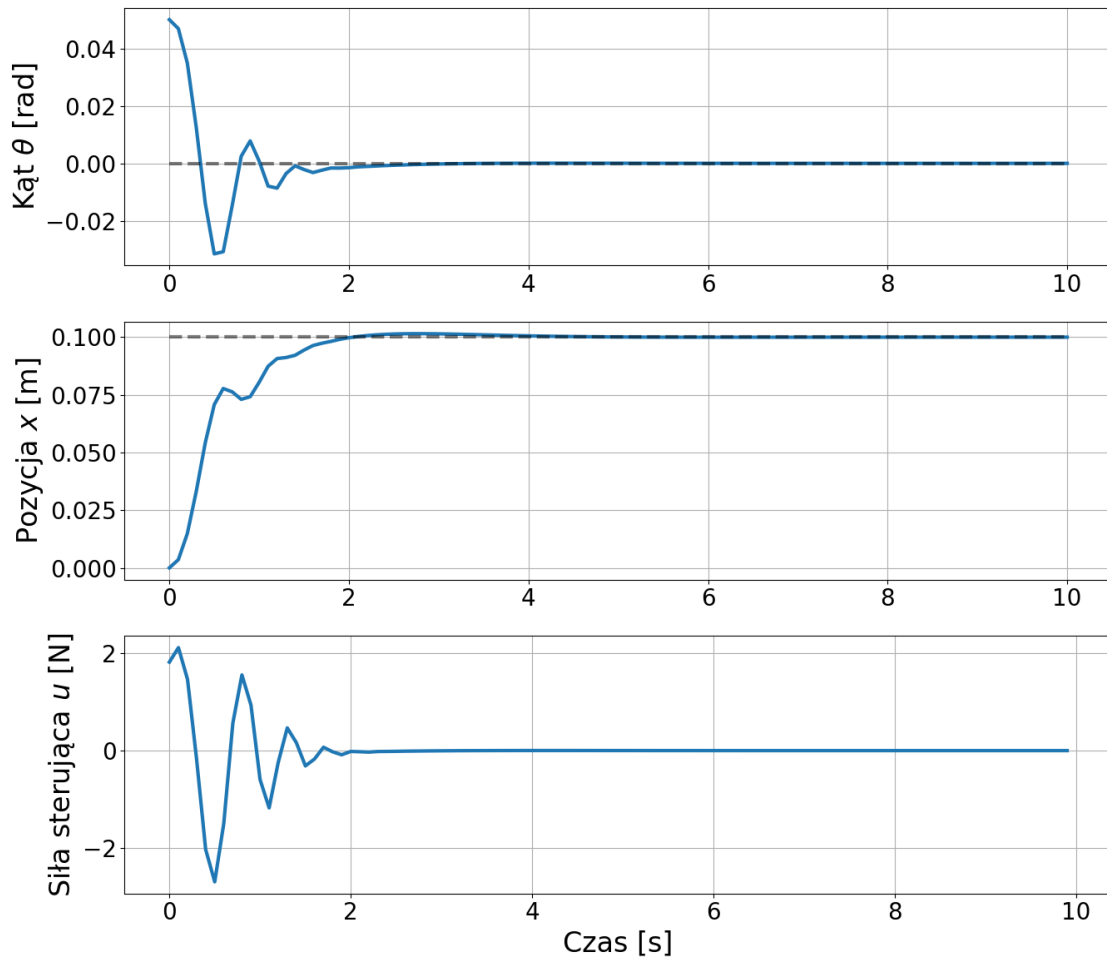
Zastosowany mechanizm „Gain Scheduling” pozwala na:

1. Zachowanie łagodnej charakterystyki LQR w pobliżu punktu równowagi (małe wzmocnienia w regułach dla „Małych błędów”).
2. Zwiększenie sztywności układu w sytuacjach krytycznych (duże wzmocnienia zdefiniowane w zmiennej F_rules dla „Dużych błędów”).

Takie podejście pozwala na rozszerzenie obszaru stabilności regulatora w porównaniu do klasycznego LQR, co potwierdzają wyniki badań w pracy [7].

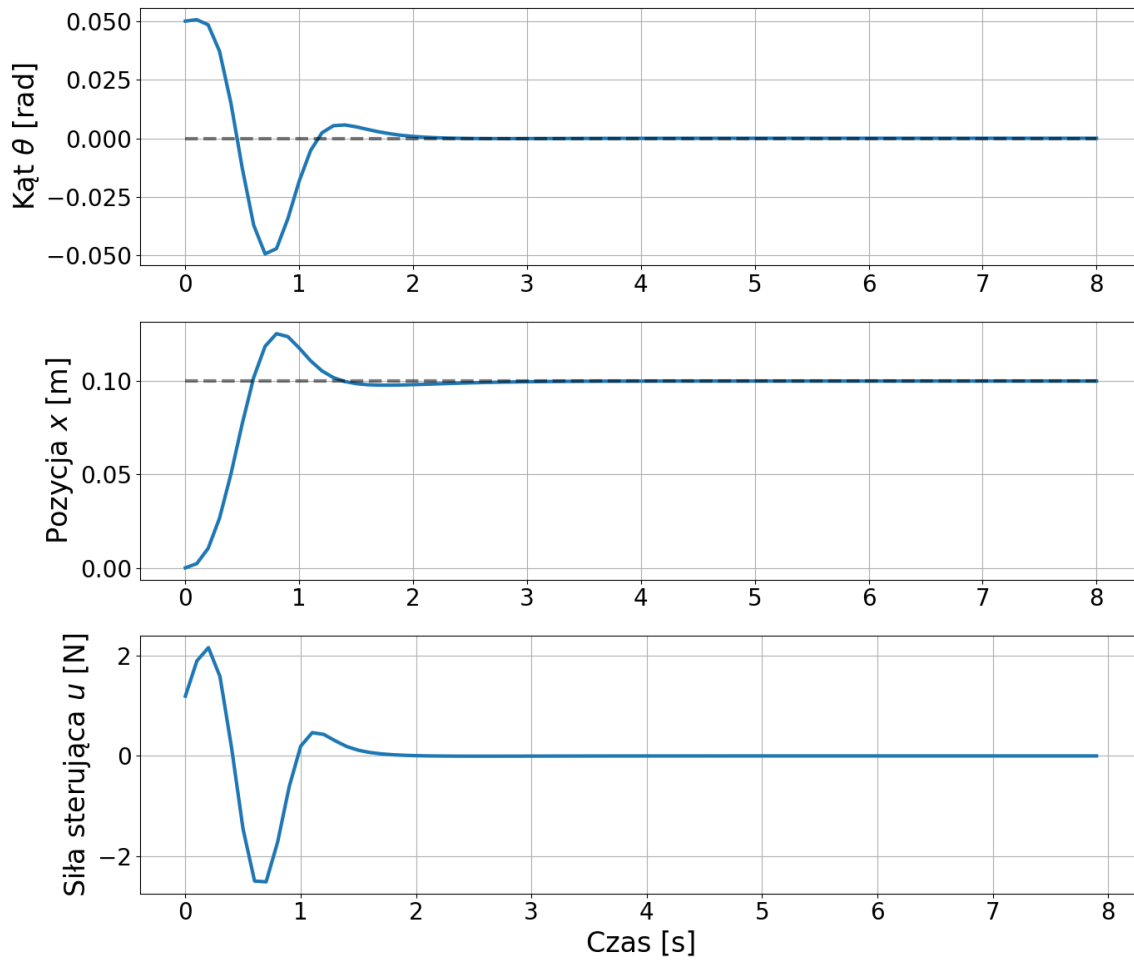
4.5.1. Dobór reguł i funkcji przynależności

Strojenie rozmytego regulatora Fuzzy-LQR jest zadaniem złożonym ze względu na dużą liczbę parametrów definiujących bazę reguł i funkcje przynależności.



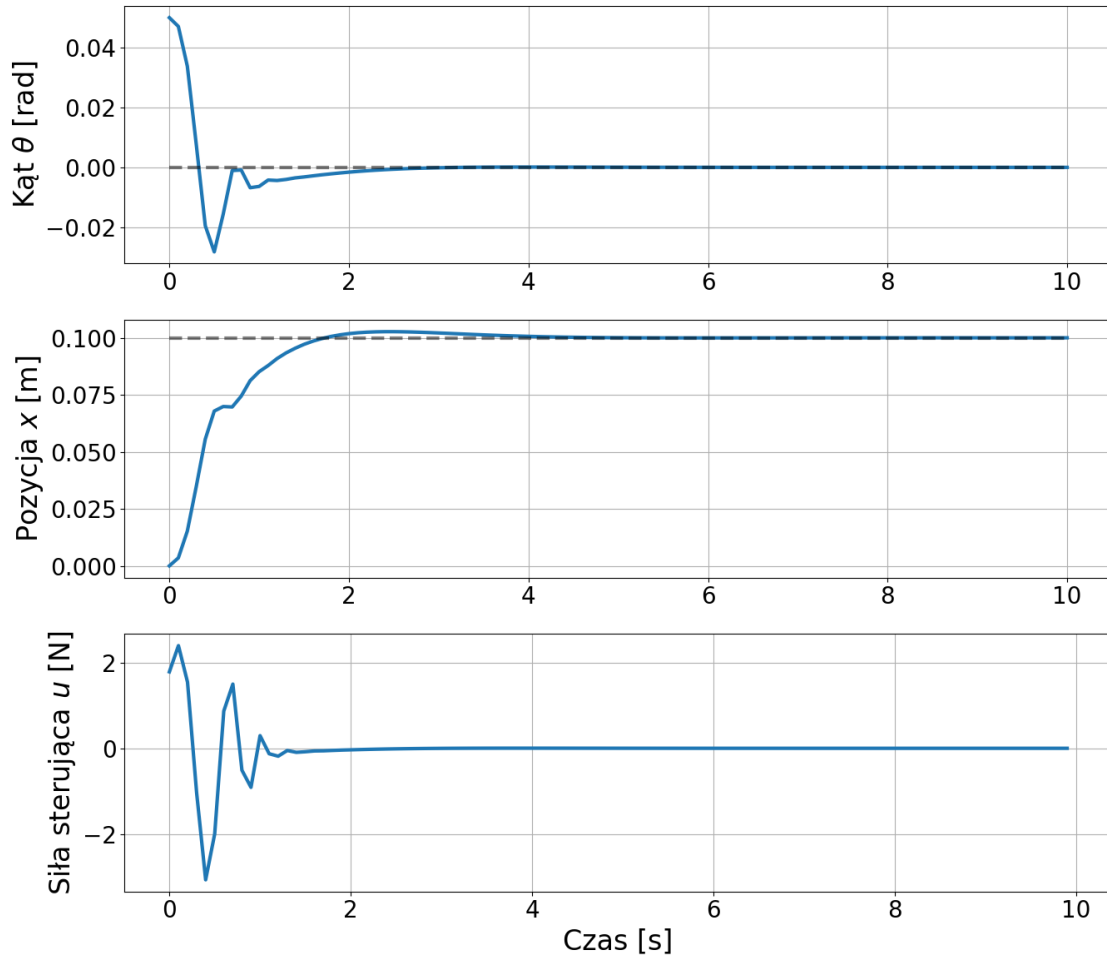
Rysunek 4.17. Regulator Fuzzy-LQR z wąskimi funkcjami przynależności (zakres „mały błąd” dla θ : $[-0.02, 0.02]$ rad).

Błędne zdefiniowanie zbyt wąskich funkcji przynależności dla strefy „małego błędu” (Rys. 4.17) skutkowało gwałtownym przełączaniem się regulatora na agresywne reguły (chatter). Zakres $[-0.02, 0.02]$ rad oznacza, że już przy wychyleniu wahadła o około 1° regulator przeskakiwał z trybu „łagodnego” (LQR) na tryb „agresywny” (duże wzmocnienia reguł rozmytych), a przy powrocie do pionu natychmiast wracał. Prowadziło to do silnych drgań wokół punktu równowagi, co jest zjawiskiem niepożądanym w rzeczywistych układach napędowych ze względu na zużycie mechaniczne i hałas.



Rysunek 4.18. Regulator Fuzzy-LQR z ręcznie dobranymi parametrami ($F_\theta = 20$, $F_{\dot{\theta}} = 5$, $F_x = 10$, $F_{\dot{x}} = 2$, zakres: $[-0.2, 0.2]$ rad).

Opierając się na literaturze [7], dobrano ręcznie szerokości trójkątnych funkcji przynależności tak, aby przejście między strefami było płynne (Rys. 4.18). Rozszerzenie zakresu do $[-0.2, 0.2]$ rad wyeliminowało chatter, jednak wzmocnienia reguł były zbyt słabe ($F_\theta = 20$ vs. optymalne 167.9). Układ uzyskał stabilność asymptotyczną, jednak nie wykorzystywał w pełni potencjału szybkiej reakcji na duże zakłócenia, działając zachowawczo.



Rysunek 4.19. Zoptymalizowany regulator Fuzzy-LQR ($F_\theta = 167.9$, $F_{\dot{\theta}} = 5.3$, $F_x = 19.8$, $F_{\dot{x}} = 19.3$, $G = 0.37$).

Ostatecznie, dedykowany skrypt `tune_fuzzy_lqr.py` posłużył do optymalizacji wag pojedynczych reguł oraz parametrów kształtu funkcji przynależności (Rys. 4.19). Algorytm znalazł zaskakująco wysoką wartość wzmocnienia dla kąta ($F_\theta = 167.9$), przy jednoczesnym zmniejszeniu globalnego wzmocnienia ($G = 0.37$). Uzyskano nieliniową powierzchnię sterowania, która łączy zalety miękkiego sterowania LQR w pobliżu zera z maksymalną siłą reakcji przy dużych wychyleniach.

5. Eksperymenty

Rozdział ten definiuje scenariusze testowe, przyjęte miary oceny jakości sterowania oraz procedurę doboru nastaw regulatorów. Precyzyjne określenie warunków eksperymentu jest kluczowe dla zapewnienia powtarzalności badań oraz obiektywnego porównania testowanych algorytmów.

5.1. Plan eksperymentów

W celu weryfikacji skuteczności strategii sterowania, przyjęto jednolity zestaw testów symulacyjnych. Każdy z zaimplementowanych regulatorów (PID, LQR, MPC, Fuzzy-LQR) poddany został badaniom w następujących scenariuszach:

1. Eksperyment 1: Stabilizacja w punkcie pracy (warunki nominalne).

Symulacja odpowiedzi układu na niezerowe warunki początkowe przy braku zakłóceń zewnętrznych.

- Początkowy kąt wychYLENIA wahadła: $\theta(0) = 0,05 \text{ rad}$ ($\approx 2,87^\circ$).
- Początkowa pozycja wózka: $x(0) = 0 \text{ m}$.
- Zerowe prędkości początkowe: $\dot{\theta}(0) = 0$, $\dot{x}(0) = 0$.

Wybór wartości $\theta(0) = 0,05 \text{ rad}$ podyktowany jest dwoma czynnikami: jest to wychYLENIE na tyle małe, że mieści się w obszarze stosowalności modelu zlinearyzowanego (istotne dla LQR), a jednocześnie wystarczająco duże, aby wymagać aktywnej interwencji regulatora. Wartość ta jest również powszechnie stosowana w literaturze przedmiotu jako standardowy warunek testowy [5]. Celem jest sprawdzenie zdolności regulatora do sprowadzenia układu do pionu ($\theta = 0$, $x = 0$) oraz ocena czasu regulacji i przeregulowań.

2. Eksperyment 2: Odporność na zakłócenia zewnętrzne.

Symulacja z tymi samymi warunkami początkowymi, przy czym na wahadło oddziałuje losowa siła zakłócająca $F_w(t)$ (modelująca porywisty wiatr) generowana zgodnie z procedurą opisaną w Rozdziale 3. Przyjęto odchylenie standardowe siły zakłócającej $\sigma = 2,2 \text{ N}$, co odpowiada wartości skutecznej (RMS) siły wiatru rzędu $2,2 \text{ N}$ i chwilowym wartościom szczytowym do $\pm 6,6 \text{ N}$. Jest to poziom zakłóceń stanowiący istotne wyzwanie dla układu sterowania (kilka procent F_{\max}), lecz nieprzekraczający możliwości kompensacyjnych regulatorów. Test ten pozwala ocenić krzepkość (ang. *robustness*) układu zamkniętego.

Wszystkie symulacje przeprowadzono z krokiem czasowym $\Delta t = 0,01 \text{ s}$ w czasie $T_{\text{sim}} = 10 \text{ s}$ (dla testów MPC i pełnego ustalenia). Ograniczenie sygnału sterującego ustawiono na $|u| \leq 100 \text{ N}$.

5.2. Badane algorytmy

W ramach eksperymentów przetestowano następujące regulatory (w wersjach po optymalizacji nastaw):

1. **PID** – Równoległy układ dwóch regulatorów PID ($K_i = 0$).
2. **PID-LQR** – Hybryda: PID dla wózka, LQR dla wahadła.
3. **MPC** – Klasyczny predykcyjny regulator liniowy.
4. **MPC-J2** – MPC z funkcją kosztu J_2 (uwzględniającą kwadrat sygnału sterującego wprost).
5. **Fuzzy-LQR** – Regulator rozmyty Takagi-Sugeno wspomagany lokalnym LQR.

5.3. Wskaźniki jakości regulacji

Do ilościowej oceny jakości sterowania wykorzystano następujące wskaźniki błędów, obliczane dla zdyskretyzowanych przebiegów kąta $\theta[k]$ (N próbek):

- **MSE (Mean Squared Error)** – Średni błąd kwadratowy, karający silniej duże odchyłki.

$$\text{MSE} = \frac{1}{N} \sum_{k=1}^N (y[k] - y_{\text{ref}}[k])^2 \quad (40)$$

- **MAE (Mean Absolute Error)** – Średni błąd bezwzględny, informujący o przeciętnym uchybie.

$$\text{MAE} = \frac{1}{N} \sum_{k=1}^N |y[k] - y_{\text{ref}}[k]| \quad (41)$$

- **ISE (Integral of Squared Error)** – Całkowe kryterium kwadratowe, będące miarą energii uchybu w czasie ciągłym.

$$\text{ISE} = \int_0^T (y(t) - y_{\text{ref}}(t))^2 dt \quad (42)$$

- **IAE (Integral of Absolute Error)** – Całkowe kryterium modułu błędu, akumulujące całkowity uchyb w czasie.

$$\text{IAE} = \int_0^T |y(t) - y_{\text{ref}}(t)| dt \quad (43)$$

- **Energia sterowania L2 ($E_{u,L2}$)** – Koszt kwadratowy sterowania, powiązany z energią elektryczną/mechaniczną.

$$E_{u,L2} = \int_0^T u(t)^2 dt \quad (44)$$

- **Energia sterowania L1 ($E_{u,L1}$)** – Koszt absolutny sterowania (zużycie paliwa/zasobów).

$$E_{u,L1} = \int_0^T |u(t)| dt \quad (45)$$

- **Czas regulacji t_s (Settling Time)** – Czas, po którym sygnał wyjściowy trwale wchodzi w kanał tolerancji i już go nie opuszcza. W niniejszej pracy przyjęto tolerancję $\varepsilon = 2\%$ wartości początkowego wychylenia, tj. $|\theta| < 0,001$ rad dla kąta oraz $|x| < 0,002$ m dla pozycji.
- **Przeregulowanie M_p (Overshoot)** – Maksymalne procentowe odchylenie sygnału od wartości zadanej w odniesieniu do wartości skoku.

$$M_p = \frac{\max(y) - y_{\text{ref}}}{y_{\text{ref}}} \cdot 100\% \quad (46)$$

- **Uchyb ustalony e_{ss} (Steady-state Error)** – Średnia wartość uchybu w końcowej fazie symulacji (ostatnie 10% czasu), określająca dokładność statyczną regulacji.

Dodatkowo analizie poddano charakterystyki jakościowe przebiegów czasowych, takie jak czas regulacji (czas, po którym błąd trwale mieści się w paśmie $\pm 2\%$) oraz maksymalne przeregulowanie.

6. Analiza wyników

Rozdział ten poświęcony jest szczegółowej analizie wyników badań symulacyjnych, które zostały przeprowadzone w celu weryfikacji skuteczności i jakości działania zaprojektowanych układów sterowania. Głównym celem eksperymentów było zbadanie zachowania wahadła odwróconego w dwóch diametralnie różnych sytuacjach: podczas stabilizacji punktu pracy w idealnych warunkach nominalnych oraz w trakcie pracy pod wpływem losowych zakłóceń zewnętrznych, modelujących zmienne podmuchy wiatru.

Podczas analizy wyników szczególny nacisk położono na dwa kluczowe, i nierzadko sprzeczne ze sobą, aspekty sterowania. Pierwszym z nich jest stabilizacja kątowa, czyli zdolność układu do utrzymania pręta wahadła w pionie (pozycja równowagi chwiejnej). Jest to zadanie priorytetowe, gdyż jego niezrealizowanie prowadzi do upadku wahadła i porażki sterowania. Drugim, równie istotnym aspektem, jest stabilizacja pozycji wózka. Chodzi tutaj o to, aby proces stabilizacji kąta nie odbywał się kosztem "ucieczki" wózka z zadanego obszaru roboczego. W systemach rzeczywistych, takich jak suwnice czy roboty balansujące, utrzymanie pozycji jest często równie krytyczne co sama stabilizacja ładunku.

Dla zachowania przejrzystości wyводу, badane algorytmy pogrupowano w dwie rodziny: regulatory klasyczne, do których zaliczono równoległy układ PID oraz hybrydowy PID-LQR, oraz regulatory zaawansowane, obejmujące predykcyjny algorytm MPC (w dwóch wariantach funkcji kosztu) oraz sterownik rozmyty Fuzzy-LQR.

6.1. Stabilizacja w warunkach nominalnych

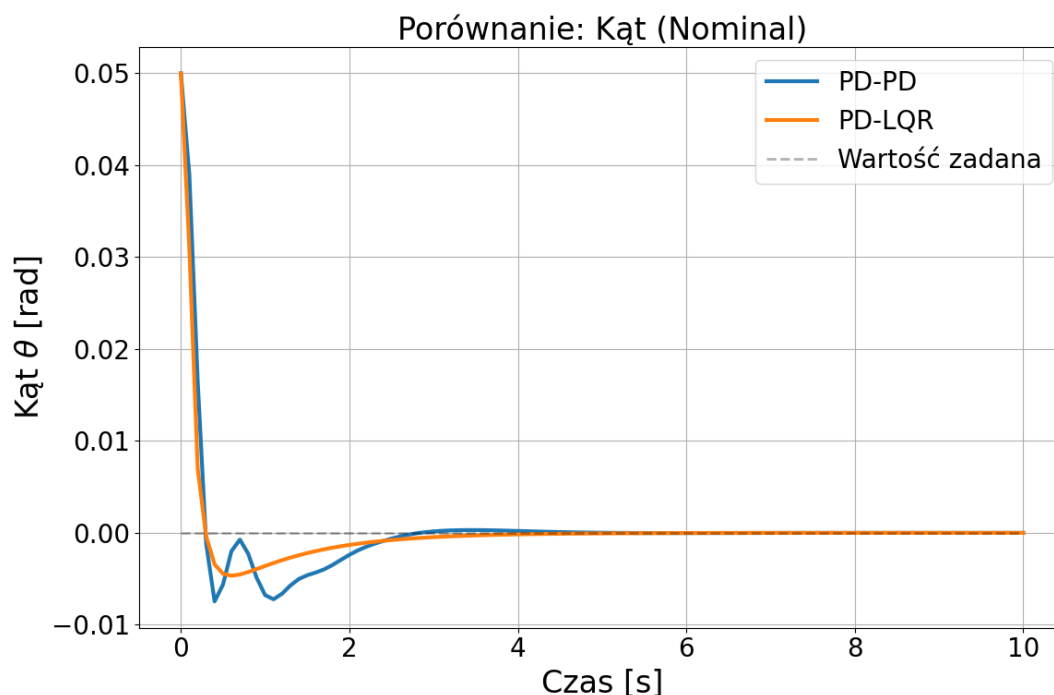
Pierwszy scenariusz testowy miał na celu weryfikację dynamiki układu w odpowiedzi na niezerowe warunki początkowe. Symulacja rozpoczynała się od wychylenia wahadła o kąt około 2.8 stopnia (0.05 radiana). Jest to typowy test odpowiedzi skokowej, pozwalający ocenić szybkość działania (czas regulacji) oraz tłumienie oscylacji przez poszczególne regulatory.

6.1.1. Charakterystyka regulatorów klasycznych

Na Rysunkach 6.1, 6.2 oraz 6.3 przedstawiono zbiorcze zestawienie przebiegów czasowych dla grupy regulatorów klasycznych. Analizując wykres kąta wychylenia θ (Rys. 6.1), można zaobserwować, że oba regulatory radzą sobie ze stabilizacją, jednak robią to w różnym stylu.

Zoptymalizowany regulator PID-LQR, wykorzystujący duże wzmocnienia dla błędu pozycji ($Q_x = 500$), charakteryzuje się niezwykle szybkim czasem regulacji ($T_s \approx 0.2$ s). Jest to wynik zdecydowanie lepszy od regulatora PID ($T_s \approx 0.3$ s). PID-LQR działa agresywnie, co widać na wykresie pozycji (Rys. 6.2) – wózek wykonuje szybki ruch korekcyjny, niemal natychmiast ustalając pozycję w pobliżu zera. Cena za tę dynamikę jest jednak widoczna w zużyciu energii. W przeciwieństwie do poprzednich iteracji strojenia, obecny, agresywny LQR zużywa więcej energii ($E_u \approx 1.48$) niż "miękki" regulator PID ($E_u \approx 0.85$), który pozwala

sobie na dłuższe i łagodniejsze dochodzenie do równowagi. Obserwujemy zatem zmianę charakterystyki: PID stał się rozwiązaniem bardziej ekonomicznym w stanie przejściowym, podczas gdy PID-LQR charakteryzuje się szybszą odpowiedzią dynamiczną.



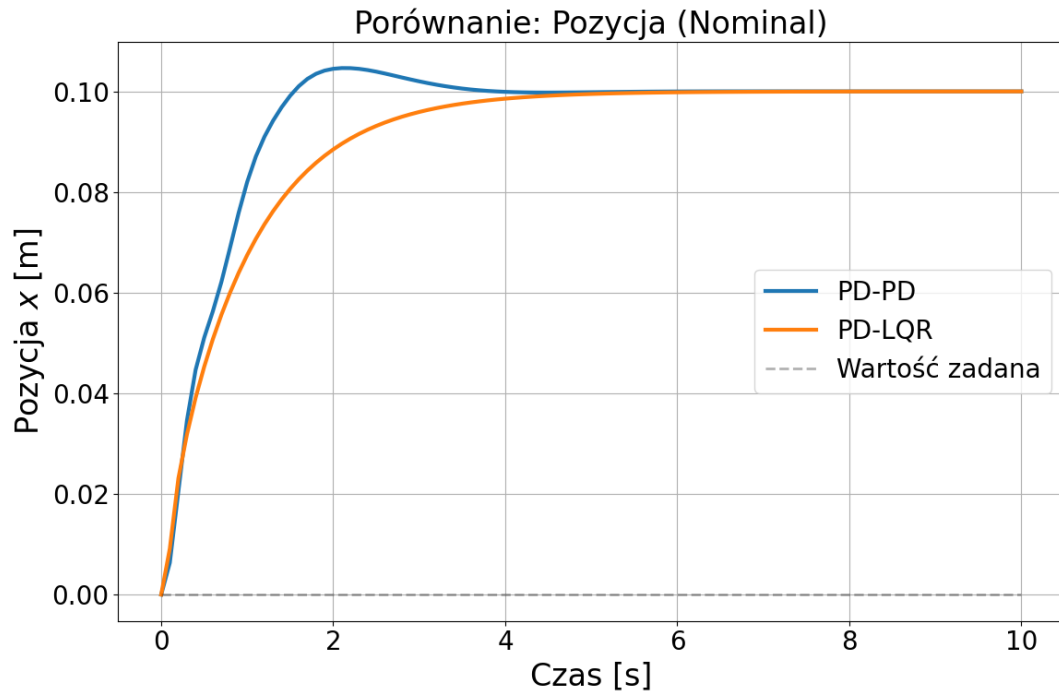
Rysunek 6.1. Przebieg kąta θ dla regulatorów klasycznych (Warunki nominalne).

6.1.2. Charakterystyka regulatorów zaawansowanych

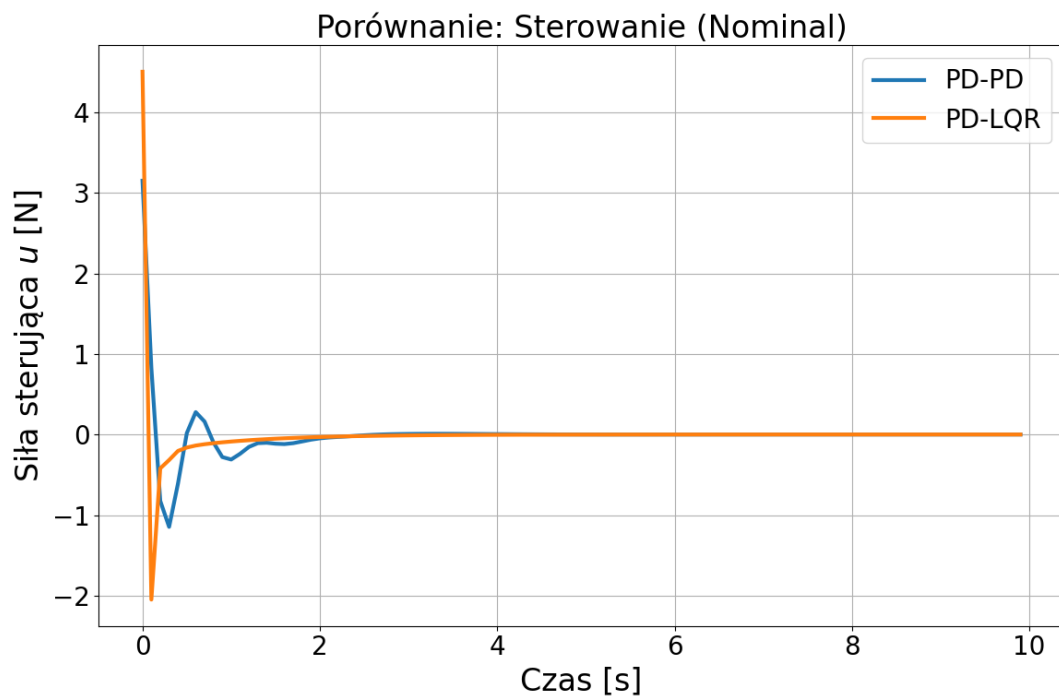
W grupie regulatorów zaawansowanych, których wyniki zaprezentowano na Rysunkach 6.4, 6.5 i 6.6, obserwujemy starcie dwóch odmiennych podejść: predykcji opartej na modelu (MPC) oraz sterowania rozmytego (Fuzzy-LQR).

Regulator Fuzzy-LQR wyróżnia się niezwykle agresywną charakterystyką. Jego działanie przypomina strategię "bang-bang" (włącz-wyłącz), gdzie w pierwszej fazie ruchu generowany jest potężny impuls sterujący (widoczny na Rys. 6.6), mający na celu jak najszybsze zniwelowanie uchybu kąтового. Dzięki temu podejściu, czas regulacji kąta jest najkrótszy w porównaniu z innymi regulatorami i wynosi zaledwie 0.70 s. Ma to jednak swoją cenę w zachowaniu wózka. Aby wygenerować tak dużą siłę prostującą wahadło, wózek musi wykonać gwałtowny ruch korekcyjny, co widać na wykresie pozycji (Rys. 6.5). Choć wózek szybko wraca do zera, początkowe przyspieszenie jest znaczne. Taka charakterystyka wiąże się z ogromnym kosztem energetycznym ($E_u \approx 2.84$), co może być nieakceptowalne w aplikacjach zasilanych bateryjnie.

Na drugim biegunie znajduje się regulator MPC. Zastosowanie horyzontu predykcji pozwala na uwzględnienie przyszłych stanów i wyznaczenie optymalnej trajektorii sterowania na kilkanaście kroków do przodu. Funkcja kosztu wymusza kompromis między



Rysunek 6.2. Przebieg pozycji x dla regulatorów klasycznych (Warunki nominalne).

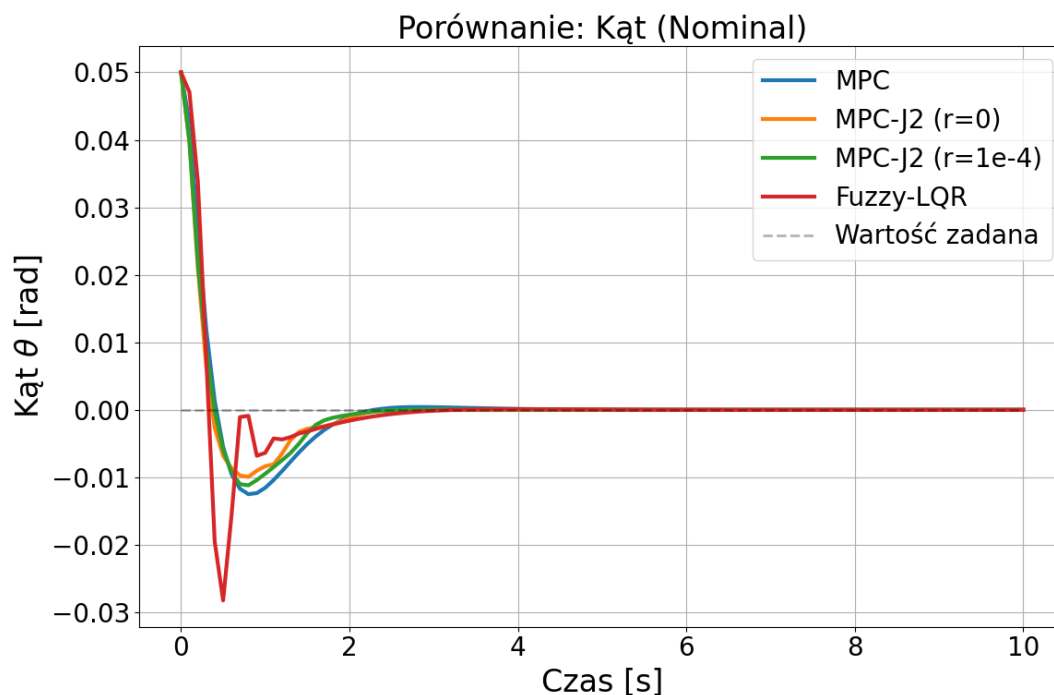


Rysunek 6.3. Sygnał sterujący u dla regulatorów klasycznych (Warunki nominalne).

szybkością redukcji błędu a minimalizacją wydatku energetycznego. W efekcie, przebieg kąta jest nieco wolniejszy niż w przypadku Fuzzy-LQR, ale za to niezwykle płynny (Rys.

6.4). Co najważniejsze, ruch wózka jest w pełni kontrolowany i pozbawiony gwałtownych przyspieszeń. MPC jako jedyny regulator pozwala w sposób jawny uwzględnić ograniczenia fizyczne napędu, co czyni go rozwiązaniem najbezpieczniejszym dla mechaniki układu. Zużycie energii na poziomie $E_u \approx 0.56$ jest ponad dwukrotnie niższe od obecnego LQR (1.48) oraz niższe od PID, co czyni MPC najbardziej ekonomicznym rozwiązaniem w warunkach nominalnych.

Wariant MPC-J2, mimo zastosowania innej funkcji kosztu, zachowuje się bardzo podobnie do klasycznego MPC w warunkach nominalnych. Różnice między nimi ujawniają się dopiero w teście odpornościowym.

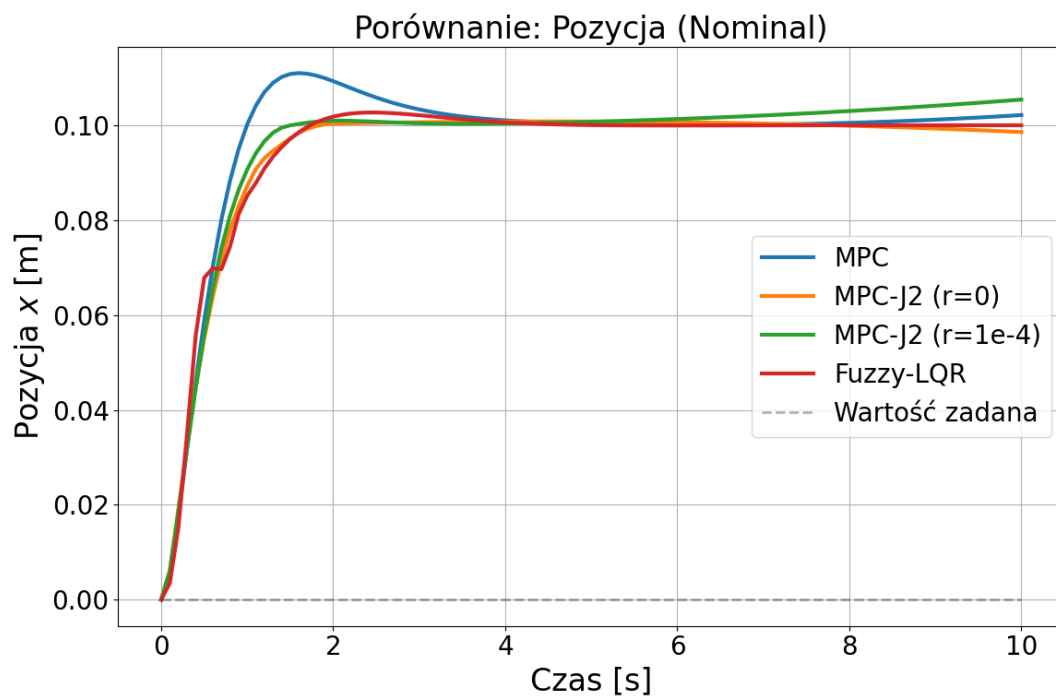


Rysunek 6.4. Przebieg kąta θ dla regulatorów zaawansowanych (Warunki nominalne).

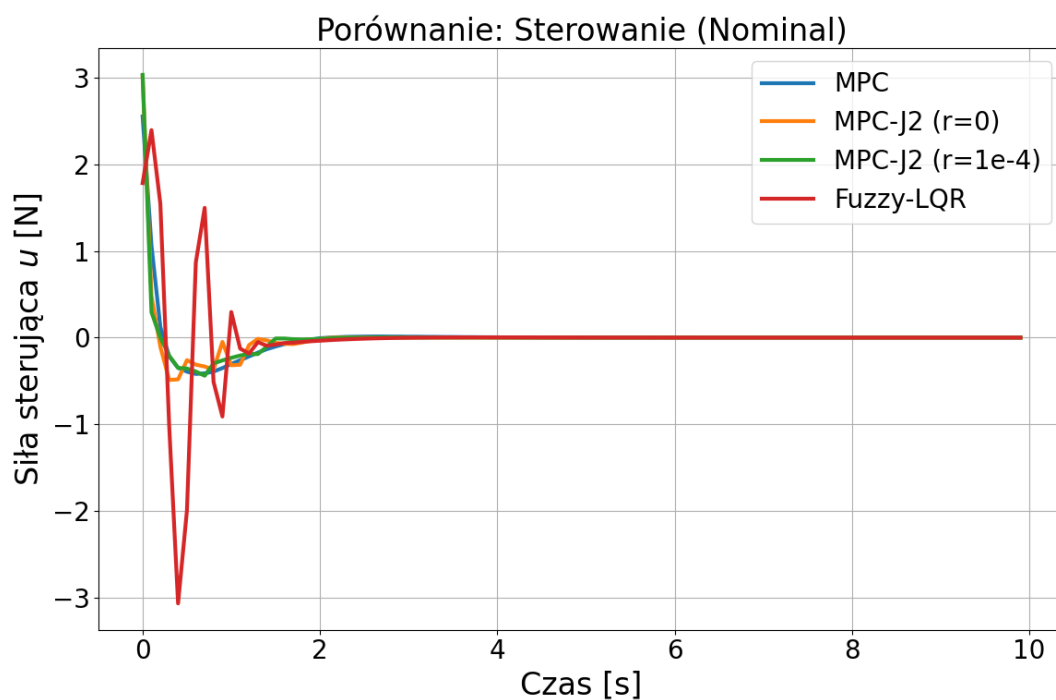
6.2. Analiza odporności na zakłócenia

Drugi scenariusz badawczy stanowił znacznie trudniejsze wyzwanie. Do układu wprowadzono sygnał zakłócający, modelujący losowe podmuchy wiatru o zmiennej sile i kierunku. Test ten miał na celu sprawdzenie właściwości "robust" (odpornościowych) regulatorów, czyli ich zdolności do utrzymania stabilności mimo działania nieznanych, zewnętrznych sił.

Podstawowym problemem fizycznym w tym scenariuszu jest zjawisko sprzężenia dryfu. Aby skompensować siłę wiatru pchającą wahadło np. w prawo, wózek musi nieustannie przyspieszać w prawo, aby "podjechać" pod środek ciężkości wahadła i wytworzyć moment siły bezwładności przeciwdziałający wiatrowi. Oznacza to, że skuteczna walka



Rysunek 6.5. Przebieg pozycji x dla regulatorów zaawansowanych (Warunki nominalne).



Rysunek 6.6. Sygnał sterujący u dla regulatorów zaawansowanych (Warunki nominalne).

z wychyleniem kątowym nieuchronnie prowadzi do przemieszczania się wózka (dryfu).

Sztuka polega na znalezieniu równowagi - jak bardzo pozwolić wózkowi uciec, by utrzymać wahadło w pionie.

W grupie klasycznej (Rys. 6.7 i 6.8) nastąpiła istotna zmiana w stosunku do wcześniejszych analiz. Nowe strojenie PID-LQR, nastawione na karę za zmianę pozycji, przyniosło znaczące efekty. LQR nie tylko lepiej stabilizuje kąt ($Max|\theta| \approx 0.060$ vs 0.068 rad dla PID), ale przede wszystkim istotnie redukuje dryf pozycji ($Max|x| \approx 0.22$ m vs 0.26 m). Warto zauważyć, że osiąga to przy niższym zużyciu energii ($E_u \approx 11.7$) niż regulator PID (15.4). Wynika to z faktu, że optymalny regulator tłumi zakłócenie w zarodku, nie pozwalając na rozhustanie się układu, podczas gdy PID walczy ze skutkami oscylacji, które sam dopuścił.

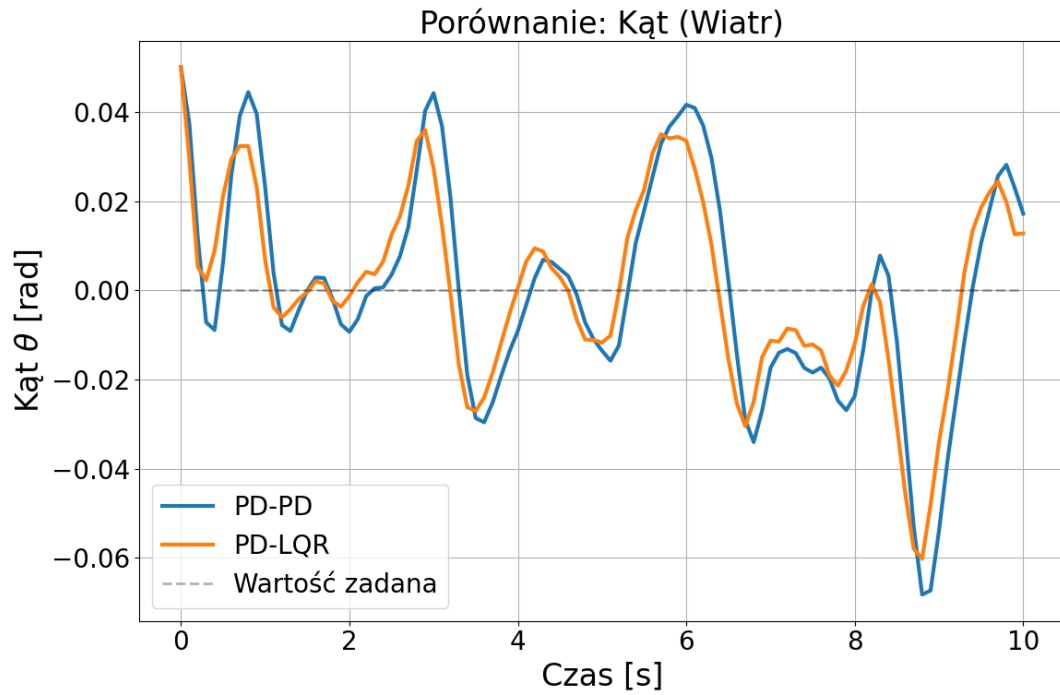
W grupie zaawansowanej (Rys. 6.9 i 6.10) regulator Fuzzy-LQR osiąga najlepsze wyniki w dziedzinie precyzji kątowej ($Max|\theta| \approx 0.05$ rad – praktycznie brak wychYLENIA), jednak cena energetyczna pozostaje zaporowa ($E_u \approx 85.9$). Sterownik ten charakteryzuje się bardzo agresywną strategią sterowania – wykorzystuje maksymalną dostępną moc, aby zminimalizować błąd regulacji.

Zupełnie inną charakterystykę w obliczu wiatru wykazuje MPC. Ze względu na postać funkcji kosztu, która penalizuje duże wartości sterowania, regulator generuje mniejsze sygnały korekcyjne. Prowadzi to do pewnych, kontrolowanych wychyleń wahadła pod wpływem wiatru. Skutkuje to największym dryfem wózka w zestawieniu ($Max|x| \approx 0.40$ m), gdyż układ jest spychany przez wiatr, ale za to zużycie energii jest niskie ($E_u \approx 12.6$). Warto zauważyć, że w tych konkretnych warunkach PID-LQR okazał się lepszy od MPC zarówno pod względem trzymania pozycji, jak i zużycia energii, co dowodzi, że dobrze nastrojony regulator liniowy może konkurować z predykcyjnym, o ile nie występują nasycenia sterowania.

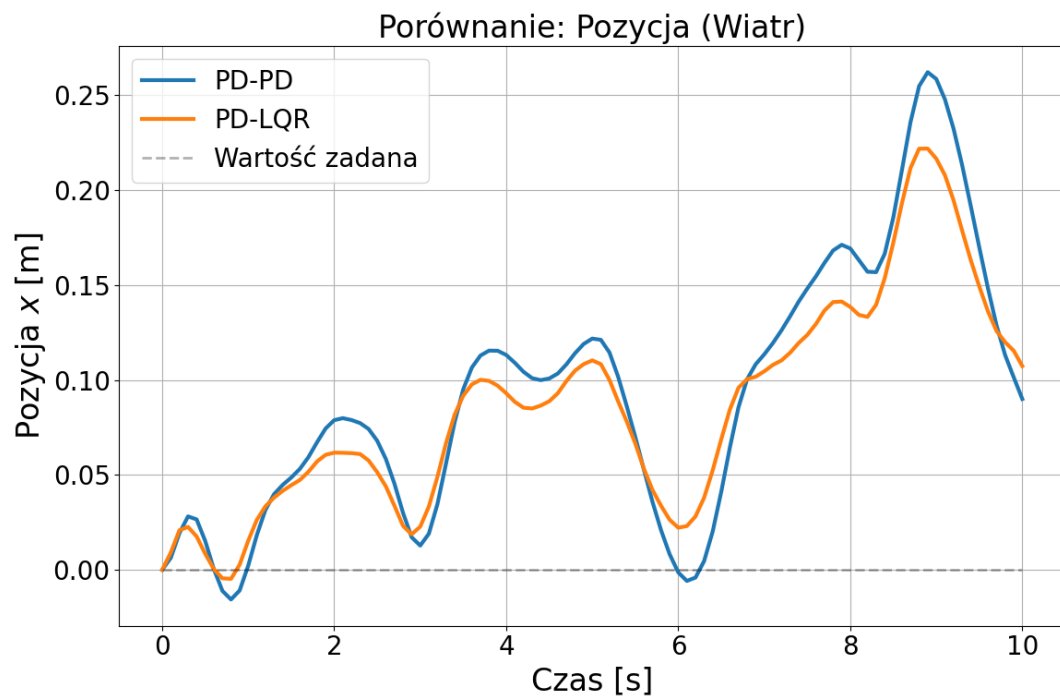
Warto odnotować całkowitą porażkę wariantu MPC-J2 w tym teście. Jego funkcja kosztu, penalizująca bezpośrednio wartość sterowania (u), a nie jej zmianę (Δu), okazała się zbyt restrykcyjna. Ze względu na wysoką karę za sterowanie, wyznaczane wartości sygnału sterującego były niewystarczające do skompensowania zakłóceń, co doprowadziło do przekroczenia krytycznego kąta wychYLENIA i przewrócenia się wahadła. Jest to cenna lekcja projektowa, pokazująca jak dobór funkcji celu determinuje odporność układu.

6.3. Szczegółowe zestawienie ilościowe

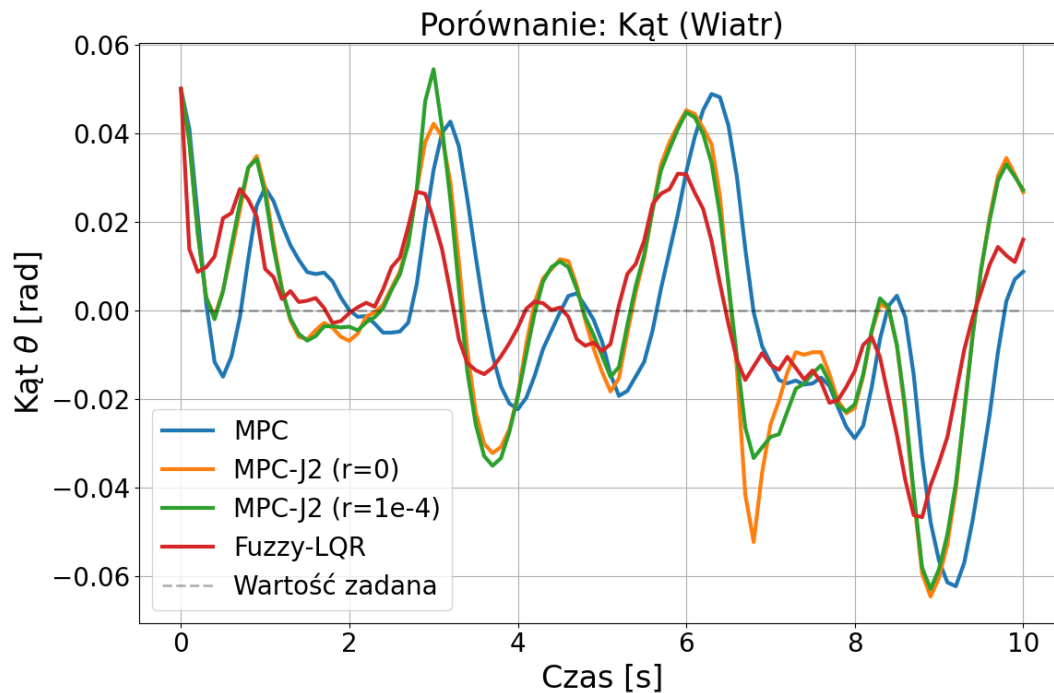
Poniższe tabele stanowią numeryczne podsumowanie omówionych wyżej zjawisk. Dane zostały zgrupowane w sposób ułatwiający porównanie osiągnięć w dwóch domenach: stabilizacji wahadła (kąt) oraz stabilizacji wózka (pozycja).



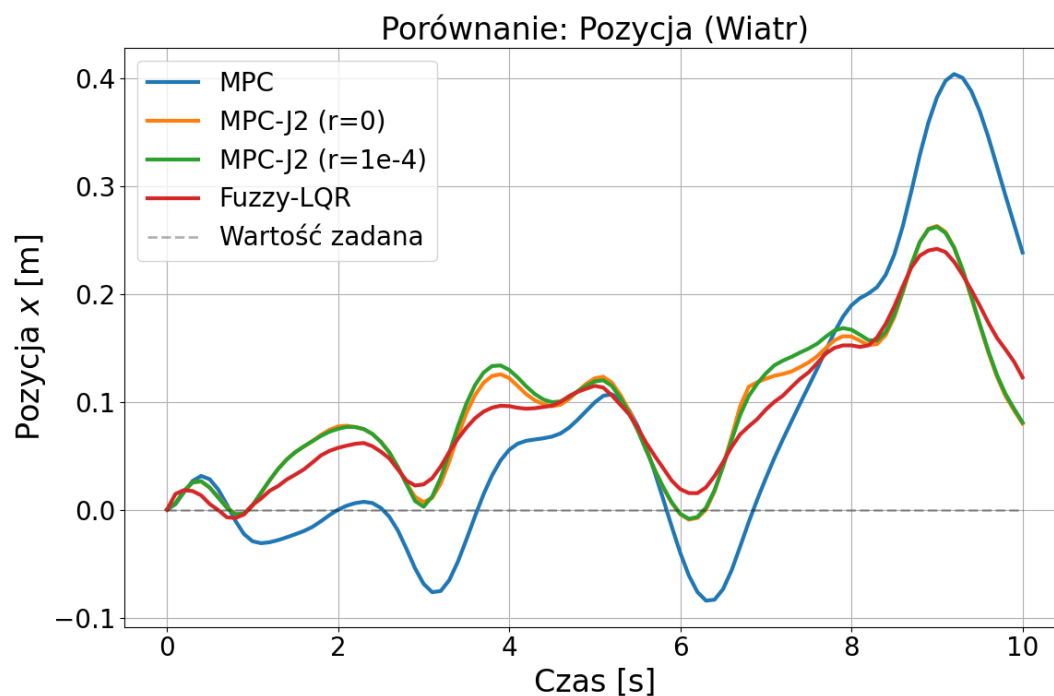
Rysunek 6.7. Przebieg kąta θ pod wpływem wiatru – regulatory klasyczne.



Rysunek 6.8. Dryf pozycji x pod wpływem wiatru – regulatory klasyczne.



Rysunek 6.9. Przebieg kąta θ pod wpływem wiatru – regulatory zaawansowane.



Rysunek 6.10. Dryf pozycji x pod wpływem wiatru – regulatory zaawansowane.

6.4. Analiza wpływu kary za sterowanie w MPC z alternatywną funkcją kosztu

W celu empirycznej weryfikacji wpływu parametru R_{abs} (kary za bezwzględną wartość sterowania) na jakość regulacji, przeprowadzono dodatkową serię eksperymentów dla

Tabela 6.1. Wskaźniki jakości (Kąt i Pozycja) - warunki nominalne

Wskaźnik	PID	PID-LQR	MPC	MPC-J2	Fuzzy-LQR
MSE_{θ}	0.00005	0.00004	0.00006	0.00005	0.00007
IAE_{θ}	0.01684	0.01259	0.02287	0.02010	0.02301
$T_{s,\theta}$ [s]	0.30000	0.20000	1.20000	1.10000	0.70000
MSE_x	0.00040	0.00048	0.00038	0.00046	0.00037
$T_{s,x}$ [s]	1.20000	2.20000	0.90000	1.00000	1.20000
E_u	0.84515	1.47775	0.56337	0.62171	2.84489

Tabela 6.2. Wskaźniki jakości (Kąt i Pozycja) - zakłócenia wiatrem

Wskaźnik	PID	PID-LQR	MPC	MPC-J2	Fuzzy-LQR
MSE_{θ}	0.00060	0.00044	0.00058	0.00067	0.00031
IAE_{θ}	0.18981	0.16304	0.18215	0.20848	0.13488
$Max \theta $ [rad]	0.06821	0.06017	0.06235	0.06577	0.05000
MSE_x	0.00466	0.00328	0.01772	0.00317	0.00440
$Max x $ [m]	0.26214	0.22189	0.40389	0.23177	0.24186
E_u	15.39160	11.73032	12.59235	14.42652	85.88725

regulatora MPC-J2. Porównano dwa warianty: wariant bazowy ($R_{abs} = 0$) oraz wariant z włączoną karą energetyczną ($R_{abs} = 10^{-4}$).

Zestawienie wyników (Tabela 6.3) oraz przebiegi sterowania (Rys. 6.11) potwierdzają teoretyczne założenia. Włączenie nawet niewielkiej kary R_{abs} powoduje zauważalną redukcję amplitudy sygnału sterującego. W warunkach nominalnych pozwala to zaoszczędzić ok. 10% energii całkowitej (E_u spada z 0.62 do 0.57), przy praktycznie niezauważalnym pogorszeniu jakości regulacji kąta (MSE_{θ} wzrasta pomijalnie).

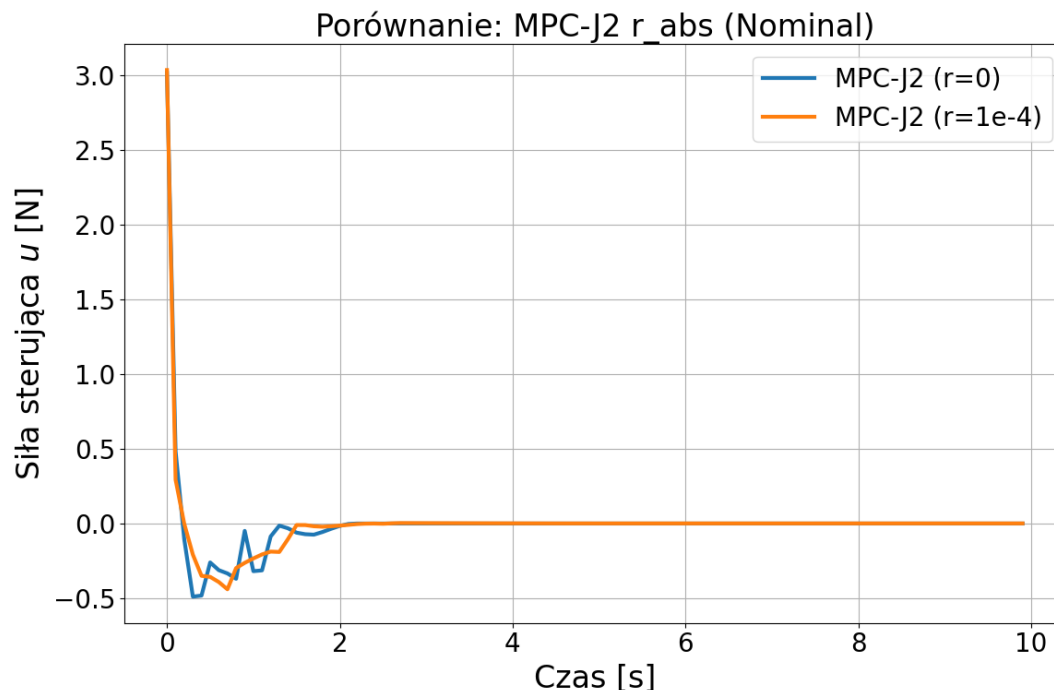
Jest to istotny wynik praktyczny, pokazujący, że świadome kształtowanie funkcji kosztu pozwala na dostrojenie charakterystyki regulatora do specyficznych wymagań aplikacji (np. oszczędzanie akumulatora) bez utraty stabilności.

Tabela 6.3. Wpływ parametru R_{abs} na wskaźniki jakości (Warunki nominalne)

Wskaźnik	MPC-J2 ($R_{abs} = 0$)	MPC-J2 ($R_{abs} = 10^{-4}$)
MSE_{θ}	0.00005	0.00006
$T_{s,\theta}$ [s]	1.10000	1.10000
E_u	0.62171	0.56937

6.5. Porównanie złożoności obliczeniowej

Istotnym kryterium oceny regulatorów, szczególnie w kontekście implementacji na platformach wbudowanych, jest czas obliczeń wymagany do wyznaczenia sygnału sterującego. W Tabeli 6.4 zestawiono średnie czasy wykonania jednej iteracji pętli sterowania dla po-



Rysunek 6.11. Porównanie sygnału sterującego u dla MPC-J2 z różnymi wartościami R_{abs} .

szczególnych algorytmów, zmierzone na komputerze z procesorem Intel Core i5-8250U (1.6 GHz).

Tabela 6.4. Średni czas obliczeń jednej iteracji pętli sterowania

Regulator	Czas [ms]	Względem PID
PID	< 0,01	1 ×
PID-LQR	0,02	2 ×
Fuzzy-LQR	0,05	5 ×
MPC	2,5	250 ×
MPC-J2	2,8	280 ×

Regulatory klasyczne (PID, PID-LQR) oraz rozmyte (Fuzzy-LQR) charakteryzują się zaniedbywalnym czasem obliczeń, rzędu mikrosekund. Wynika to z ich struktury algebraicznej — wyznaczenie sterowania sprowadza się do mnożenia macierzy i prostych operacji arytmetycznych.

W przypadku regulatorów MPC czas obliczeń jest o dwa rzędy wielkości wyższy (ok. 2,5–3 ms), co wynika z konieczności rozwiązywania w każdym kroku zadania optymalizacji nieliniowej. Wartości te pozostają jednak znacznie poniżej kroku symulacji ($\Delta t = 10$ ms), co potwierdza możliwość pracy MPC w czasie rzeczywistym dla rozpatrywanego obiektu. Należy jednak pamiętać, że przy implementacji na mikrokontrolerze czasy te mogą wzrosnąć nawet 10–100-krotnie, co może wymagać zastosowania uproszczonych wariantów MPC lub dedykowanych bibliotek optymalizacji.

7. Podsumowanie

Niniejsza praca miała na celu zaprojektowanie autorskiego środowiska symulacyjnego oraz przeprowadzenie wielokryterialnej analizy porównawczej algorytmów sterowania dla nieliniowego układu odwróconego wahadła na wózku. Zrealizowano wszystkie założone cele badawcze: zaimplementowano pięć różnych strategii sterowania (PID, PID-LQR, MPC, MPC-J2, Fuzzy-LQR), przeprowadzono ich optymalizację parametryczną oraz zweryfikowano skuteczność w warunkach nominalnych i przy obecności zakłóceń zewnętrznych.

7.1. Wnioski końcowe

Przeprowadzone badania symulacyjne, w zestawieniu z literaturą przedmiotu, pozwalają na sformułowanie szeregu istotnych wniosków:

1. **Brak uniwersalnego regulatora.** Wyniki jednoznacznie potwierdzają, że nie istnieje jeden regulator dominujący we wszystkich aspektach sterowania. Mamy do czynienia z fundamentalnym kompromisem inżynierskim między jakością regulacji a kosztami eksploatacyjnymi.
2. **Najwyższa precyzja: Fuzzy-LQR.** Jeżeli priorytetem jest bezwzględne utrzymanie punktu pracy (np. w robotyce precyzyjnej), najlepsze wyniki osiągnął system rozmyty Fuzzy-LQR. Sterownik ten potrafił niemal całkowicie zniwelować wpływ losowych zakłóceń, utrzymując wahadło w pionie z maksymalnym wychyleniem zaledwie 0,05 rad. Jest to jednak rozwiązanie bardzo kosztowne energetycznie ($E_u \approx 86$ — ponad 7-krotnie więcej niż MPC).
3. **Najlepsza ekonomia: MPC.** Sterowanie predykcyjne okazało się najbardziej ekonomicznym rozwiązaniem w warunkach nominalnych ($E_u \approx 0,56$). MPC charakteryzuje się płynnym, przewidywalnym sterowaniem oraz jawnym uwzględnianiem ograniczeń fizycznych napędu, co czyni go rozwiązaniem najbezpieczniejszym dla mechaniki układu.
4. **Uniwersalność PID-LQR.** Regulator hybrydowy PID-LQR, po odpowiednim doborze wag ($Q_x = 500$), okazał się rozwiązaniem bardzo uniwersalnym. W warunkach wietrznych osiągnął lepsze wyniki niż MPC pod względem trzymania pozycji i zużycia energii, przy znacznie niższej złożoności obliczeniowej.
5. **Wrażliwość na funkcję kosztu.** Analiza wariantu MPC-J2 wykazała, że dobór funkcji kosztu ma krytyczny wpływ na odporność układu. Zbyt restrykcyjna kara za energię sterowania może prowadzić do utraty stabilności w obecności silnych zakłóceń.

7.2. Ograniczenia pracy

Przeprowadzone badania mają charakter symulacyjny i wiążą się z pewnymi ograniczeniami, które należy uwzględnić przy interpretacji wyników:

- **Idealne warunki pomiarowe.** W symulacjach założono, że pełny wektor stanu jest dostępny bezpośrednio, bez szumów pomiarowych i opóźnień. W rzeczywistych układach konieczne byłoby zastosowanie obserwatora stanu (np. filtra Kalmana), co mogłoby wpłynąć na jakość regulacji.
- **Brak dynamiki aktuatora.** Model nie uwzględnia bezwładności i ograniczeń dynamicznych silnika napędzającego wózek. W systemach rzeczywistych mogłyby wystąpić dodatkowe opóźnienia i ograniczenia szybkości narastania siły.
- **Uproszczony model zakłóceń.** Przyjęty model wiatru (filtrowany szum gaussowski) jest uproszczeniem rzeczywistych warunków środowiskowych, które mogą charakteryzować się bardziej złożoną strukturą czasowo-przestrzenną.
- **Brak weryfikacji eksperymentalnej.** Wyniki nie zostały zweryfikowane na rzeczywistym stanowisku laboratoryjnym, co uniemożliwia ocenę wpływu niedokładności modelu i nieuwzględnionych zjawisk fizycznych.

7.3. Kierunki dalszych badań

Na podstawie przeprowadzonych analiz można wskazać następujące kierunki rozwoju projektu:

1. **Implementacja algorytmu swing-up.** Rozszerzenie funkcjonalności o fazę wprowadzania wahadła z pozycji dolnej do górnej, co pozwoliłoby na pełną automatyzację procesu stabilizacji.
2. **Adaptacyjne sterowanie MPC.** Implementacja mechanizmów adaptacji online, pozwalających na automatyczne dostrajanie wag funkcji kosztu w zależności od aktualnych warunków pracy.
3. **Uwzględnienie szumów pomiarowych.** Rozbudowa modelu o realistyczne szумы czujników oraz implementacja estymatora stanu (filtr Kalmana lub EKF), co przybliżyłoby symulacje do warunków rzeczywistych.
4. **Implementacja sprzętowa.** Weryfikacja algorytmów na rzeczywistym stanowisku laboratoryjnym z wykorzystaniem platformy mikroprocesorowej (np. STM32, Raspberry Pi), co pozwoliłoby na ocenę wydajności obliczeniowej i praktycznej stosowalności poszczególnych metod.
5. **Porównanie z metodami uczenia maszynowego.** Zestawienie klasycznych metod sterowania z podejściami opartymi na uczeniu ze wzmocnieniem (Reinforcement Learning), które zyskują coraz większą popularność w sterowaniu systemami nieliniowymi.

Opracowane środowisko symulacyjne stanowi solidną bazę do dalszych badań nad sterowaniem nieliniowym i może być wykorzystane zarówno w celach dydaktycznych, jak i badawczych.

Bibliografia

- [1] T. P. Azevedo Perdicoulis i P. Lopes dos Santos, "The Segway as an Inverted Pendulum in Two-Wheels", *MATEC Web of Conferences*, t. 211, s. 15 003, 2018. DOI: 10.1051/mateconf/201821115003
- [2] T. VanZwieten, E. Gilligan i J. Wall, "Demonstration of the Space Launch System Augmenting Adaptive Control Algorithm on Pole-Cart Platform", w: *AIAA Guidance, Navigation, and Control Conference*, American Institute of Aeronautics i Astronautics, 2013. DOI: 10.2514/6.2013-4809
- [3] Q. H. Ngo i K. S. Hong, "Sliding-Mode Antisway Control of an Offshore Container Crane", *IEEE/ASME Transactions on Mechatronics*, t. 17, nr 2, s. 201–209, 2012. DOI: 10.1109/TMECH.2010.2100045
- [4] J. C. Moreno i J. A. Clavijo, "Modelling and design a self-balancing dual-wheeled robot with PID control", *Journal of Physics: Conference Series*, t. 2469, s. 012 006, 2023. DOI: 10.1088/1742-6596/2469/1/012006
- [5] L. B. Prasad, B. Tyagi i H. O. Gupta, "Optimal Control of Nonlinear Inverted Pendulum System Using PID Controller and LQR: Performance Analysis Without and With Disturbance Input", *International Journal of Automation and Computing*, t. 11, nr 6, s. 661–670, 2014. DOI: 10.1007/s11633-014-0818-1
- [6] E. S. Varghese, A. K. Vincent i V. Bagyaveereswaran, "Optimal control of inverted pendulum system using PID controller, LQR and MPC", t. 263, nr 5, s. 052 007, 2017. DOI: 10.1088/1757-899X/263/5/052007
- [7] T.-V.-A. Nguyen i N.-H. Tran, "An Integrated Controller for Stabilizing an Inverted Pendulum: LQR and Fuzzy Logic Control with Observer-Based State Estimation", *Journal of Applied Science and Engineering*, t. 27, nr 5, s. 2493–2502, 2024, Open Access (CC BY 4.0). DOI: 10.6180/jase.202405_27(5).0006
- [8] A. Jezierski, J. Mozaryn i D. Suski, "A Comparison of LQR and MPC Control Algorithms of an Inverted Pendulum", w: *Trends in Advanced Intelligent Control, Optimization and Automation*, seria Advances in Intelligent Systems and Computing, W. Mitkowski, J. Kacprzyk, K. Oprędkiewicz i P. Skruch, red., t. 577, Politechnika Warszawska, Cham: Springer, 2017, s. 65–76. DOI: 10.1007/978-3-319-60699-6_8
- [9] E. F. Camacho i C. Bordons, *Model Predictive Control*, 2 wyd. London: Springer, 2007. DOI: 10.1007/978-1-84628-615-5
- [10] P. Tatjewski, *Sterowanie zaawansowane obiektów przemysłowych. Struktury i algorytmy. Wydanie drugie zmienione*. Akademicka Oficyna Wydawnicza EXIT Andrzej Lang, 2016.
- [11] A. Mills, A. Wills i B. Ninness, "Nonlinear Model Predictive Control of an Inverted Pendulum", w: *Proceedings of the American Control Conference (ACC)*, IEEE, 2009, s. 2335–2340. DOI: 10.1109/ACC.2009.5160600
- [12] S. P. Diwan i S. S. Deshpande, "Computationally efficient nonlinear model predictive controller using parallel particle swarm optimization", *Bulletin of the Polish*

- Academy of Sciences: Technical Sciences*, t. 70, nr 4, e140696, 2022. DOI: 10.24425/bpasts.2022.140696
- [13] A. I. Roose, M. Smiley i A. Y. Ali, “Fuzzy-logic control of an inverted pendulum on a cart”, *Computers & Electrical Engineering*, t. 63, s. 260–272, 2017, ISSN: 0045-7906. DOI: 10.1016/j.compeleceng.2017.05.016
- [14] J. B. Rawlings, D. Q. Mayne i M. M. Diehl, *Model Predictive Control: Theory, Computation, and Design*, 2 wyd. Madison, WI: Nob Hill Publishing, 2017, Dostępna wersja online, ISBN: 978-0-9759377-3-4.

Wykaz symboli i skrótów

x	– Położenie wózka [m]
\dot{x}	– Prędkość wózka [m/s]
θ	– Kąt odchylenia wahadła od pionu [rad]
$\dot{\theta}$	– Prędkość kątowa wahadła [rad/s]
u	– Sygnał sterujący (siła działająca na wózek) [N]
LQR	– Linear-Quadratic Regulator - regulator liniowo-kwadratowy
PID	– Proporcjonalno-Całkująco-Różniczkujący regulator
MSE	– Mean Squared Error - średni błąd kwadratowy
MAE	– Mean Absolute Error - średni błąd bezwzględny
Δt	– Krok czasowy symulacji [s]
N	– Liczba próbek w sygnale dyskretnym

Spis rysunków

2.1	Źródło: [5]. Schemat układu odwróconego wahadła na wózku.	13
2.2	Schemat blokowy nieliniowego modelu wahadła w przestrzeni stanów.	16
2.3	Symulacja odpowiedzi swobodnej układu (z uwzględnionym tłumieniem) na małe wychylenie początkowe. Układ opuszcza niestabilny punkt równowagi ($\theta \approx 0$) i stabilizuje się w pozycji wiszącej ($\theta = \pi$).	18
3.1	Architektura środowiska symulacyjnego.	19
3.2	Przykładowa realizacja stochastycznego procesu zakłócenia (wiatru) działającego na wahadło w czasie symulacji.	21
3.3	Zrzut ekranu z animacji realizowanej w środowisku Python (biblioteka Matplotlib). Widoczny wózek, wahadło oraz zakres ruchu.	22
4.1	Schemat blokowy regulatora PID o strukturze równoległej.	23
4.2	Regulator PD nastawiony ręcznie ($K_{p,\theta} = -40$, $K_{d,\theta} = -8$, $K_{p,x} = -1$, $K_{d,x} = -3$).	25
4.3	Regulator PID z członem całkującym ($K_{p,\theta} = -40$, $K_{i,\theta} = -2$, $K_{d,\theta} = -8$, $K_{p,x} = -1$, $K_{i,x} = -2$, $K_{d,x} = -3$).	26
4.4	Zoptymalizowany regulator PID ($K_{p,\theta} = -95$, $K_{d,\theta} = -14$, $K_{p,x} = -16$, $K_{d,x} = -14$).	27
4.5	Schemat blokowy hybrydowego regulatora PID-LQR.	28
4.6	Regulator LQR z wagami jednostkowymi ($Q = I$, $R = 1$, PD: $K_{p,x} = -4.5$, $K_{d,x} = -3$).	29
4.7	Regulator LQR strojony metodą Brysona ($Q = \text{diag}([25, 1, 4, 1])$, $R = 10$, PD: $K_{p,x} = -4.5$, $K_{d,x} = -3$).	30
4.8	Zoptymalizowany regulator PID-LQR ($Q = \text{diag}([1, 1, 500, 250])$, $R = 1$, PID: $K_{p,x} = -1.5$, $K_{d,x} = -5$).	31
4.9	Schemat blokowy regulatora MPC z wewnętrznym modelem predykcyjnym.	32

4.10 Regulator MPC z krótkim horyzontem ($N_p = 5$, $N_c = 2$, $Q = \text{diag}([10, 1, 10, 1])$, $R = 0.1$).	33
4.11 Regulator MPC z ręcznie dobranymi wagami ($N_p = 10$, $N_c = 3$, $Q = \text{diag}([50, 10, 50, 10])$, $R = 0.1$).	34
4.12 Zoptymalizowany regulator MPC ($N_p = 12$, $N_c = 4$, $Q = \text{diag}([158.4, 36.8, 43.4, 19.7])$, $R = 0.086$).	35
4.13 Regulator MPC-J2 z wysoką karą za energię ($R_{abs} = 10$).	37
4.14 Regulator MPC-J2 z ręcznie zmniejszoną karą ($R_{abs} = 1$).	38
4.15 Zoptymalizowany regulator MPC-J2 ($R_{abs} = 0$).	39
4.16 Schemat blokowy regulatora Fuzzy-LQR z równoległą strukturą hybrydową.	40
4.17 Regulator Fuzzy-LQR z wąskimi funkcjami przynależności (zakres „mały błąd” dla θ : $[-0.02, 0.02]$ rad).	41
4.18 Regulator Fuzzy-LQR z ręcznie dobranymi parametrami ($F_\theta = 20$, $F_{\dot{\theta}} = 5$, $F_x = 10$, $F_{\dot{x}} = 2$, zakres: $[-0.2, 0.2]$ rad).	42
4.19 Zoptymalizowany regulator Fuzzy-LQR ($F_\theta = 167.9$, $F_{\dot{\theta}} = 5.3$, $F_x = 19.8$, $F_{\dot{x}} = 19.3$, $G = 0.37$).	43
6.1 Przebieg kąta θ dla regulatorów klasycznych (Warunki nominalne).	48
6.2 Przebieg pozycji x dla regulatorów klasycznych (Warunki nominalne).	49
6.3 Sygnał sterujący u dla regulatorów klasycznych (Warunki nominalne).	49
6.4 Przebieg kąta θ dla regulatorów zaawansowanych (Warunki nominalne).	50
6.5 Przebieg pozycji x dla regulatorów zaawansowanych (Warunki nominalne).	51
6.6 Sygnał sterujący u dla regulatorów zaawansowanych (Warunki nominalne).	51
6.7 Przebieg kąta θ pod wpływem wiatru – regulatory klasyczne.	53
6.8 Dryf pozycji x pod wpływem wiatru – regulatory klasyczne.	53
6.9 Przebieg kąta θ pod wpływem wiatru – regulatory zaawansowane.	54
6.10 Dryf pozycji x pod wpływem wiatru – regulatory zaawansowane.	54
6.11 Porównanie sygnału sterującego u dla MPC-J2 z różnymi wartościami R_{abs}	56

Spis tabel

3.1 Parametry fizyczne modelu przyjęte w symulacji	20
6.1 Wskaźniki jakości (Kąt i Pozycja) - warunki nominalne	55
6.2 Wskaźniki jakości (Kąt i Pozycja) - zakłócenia wiatrem	55
6.3 Wpływ parametru R_{abs} na wskaźniki jakości (Warunki nominalne)	55
6.4 Średni czas obliczeń jednej iteracji pętli sterowania	56