

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 2025



# **Efektywny układ stabilizacji odwróconego wahadła na wózku**

## **Streszczenie.**

TODO

**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.**

TODO

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

# Spis treści

<b>1. Wstęp</b>	7
1.1. Cel i zakres pracy	8
1.2. Przegląd literatury	9
1.3. Układ pracy	10
<b>2. Model matematyczny układu</b>	12
2.1. Opis fizyczny i założenia upraszczające	12
2.2. Analiza kinematyczna	12
2.3. Równania dynamiki układu	13
2.3.1. Bilans sił w kierunku poziomym	13
2.3.2. Bilans momentów sił	14
2.4. Nieliniowy model w przestrzeni stanów	14
2.5. Linearyzacja modelu w punkcie pracy	15
2.6. Analiza zachowania układu w pętli otwartej	16
<b>3. Środowisko symulacyjne i implementacja</b>	17
3.1. Narzędzia programistyczne	17
3.2. Konfiguracja symulacji	17
3.3. Modelowanie zakłóceń	18
3.4. Wizualizacja i animacja	19
<b>4. Algorytmy sterowania</b>	21
4.1. Równoległy regulator PD-PD	21
4.1.1. Proces doboru nastaw oraz analiza PID	22
4.2. Układ hybrydowy PD-LQR	25
4.2.1. Dobór wag macierzy Q i R	26
4.3. Nieliniowe sterowanie predykcyjne (MPC)	29
4.3.1. Dobór horyzontu i wag funkcji celu	30
4.4. MPC z rozszerzonym wskaźnikiem jakości (MPC-J2)	31
4.4.1. Analiza wpływu kary za energię	32
4.5. Regulator rozmyty wspomagany LQR (Fuzzy-LQR)	33
4.5.1. Dobór reguł i funkcji przynależności	33
<b>5. Eksperymenty</b>	35
5.1. Plan eksperymentów	35
5.2. Badane algorytmy	35
5.3. Wskaźniki jakości regulacji	36
<b>6. Analiza wyników eksperymentalnych</b>	37
6.1. Eksperyment 1: Stabilizacja w warunkach nominalnych	37
6.2. Eksperyment 2: Odporność na wiatr	38
6.3. Analiza ilościowa – zestawienie wskaźników	38

6.4. Dyskusja wyników . . . . .	39
<b>Wykaz symboli i skrótów . . . . .</b>	<b>41</b>
<b>Spis rysunków . . . . .</b>	<b>41</b>
<b>Spis tabel . . . . .</b>	<b>42</b>

# 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 **segway2018**;
- sterowanie rakietami nośnymi i pociskami w fazie startu, gdzie wektor ciągu pełni rolę siły napędowej wózka, a bezwładność korpusu odpowiada dynamice wahadła **nasa2013**;
- sterowanie ruchem ramion manipulatorów i suwnic, w których kluczowe jest tłumienie oscylacji przenoszonego ładunku **crane2012**;
- równoważenie platform i pojazdów samobalansujących, wymagające ciągłej korekty siły napędowej względem położenia środka masy **moreno2023**.

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 zestaw wymagań funkcjonalnych i нефункциональных, które musi spełniać opracowany system:

Aby zrealizować ten cel, zdefiniowano szereg wymagań funkcjonalnych i нефункциональных 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 нефункциональных 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),
- inteligentne (sterowanie rozmyte).

Wyróżnikiem pracy jest to, że 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 trudnymi 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 oceniono ekonomię sterowania, wyznaczając koszt energetyczny poprzez normy  $L_1$  i  $L_2$  sygnału sterującego. Całość uzupełnia analiza odporności układu na zakłócenia (szumy, siły zewnętrzne), wyrażona m.in. wskaźnikiem SNR. 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 **art1** 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 **art2**. 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 **art3** 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. **jezierski2017** 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 **camacho2007** oraz Tatjewskiego **tatjewski\_pl**. 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 **mills2009**, **diwan2022**. Natomiast w obszarze sterowania rozmytego cennym źródłem wiedzy metodycznej są opracowania dotyczące modeli Takagi–Sugeno i ich porównań z podejściami klasycznymi **roose2017**.

Przeprowadzona analiza literatury wskazuje na ewolucję podejść sterowania: od klasycznych paradygmatów PID i LQR **art1**, przez ujęcia predykcyjne **art2**, **tatjewski\_pl**, aż po zaawansowane metody hybrydowe i inteligentne **art3**.

### 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 sterowa-

nia. 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 **art1**.

### 2.1. Opis fizyczny i założenia upraszczające

Rozważany obiekt sterowania należy do klasy mechanicznych układów podwzbudnych (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 **art1**:

- $M$  — 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 [ $\text{m/s}^2$ ],
- $F(t)$  — 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 **art1**:

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  $x - y$ .

TODO - zdjęcie jakies

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 **art1**:

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

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

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

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 **art1**.

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

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

## 2. Model matematyczny układu

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 **art1**, 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:

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

Po wymnożeniu nawiasów składniki zawierające  $\dot{\theta}^2$  wzajemnie się redukują. Wykorzystując jedynekę trygonometryczną  $\cos^2\theta + \sin^2\theta = 1$  oraz dzieląc całe równanie przez  $l$ , otrzymujemy ostateczne drugie równanie dynamiki **art1**:

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

### 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 **art1**:

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

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

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

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

## 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 **art1**:

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

Macierz sterowania  $\mathbf{B}$  oraz macierz zakłóceń  $\mathbf{B}_{\text{dist}}$  przyjmują postać:

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

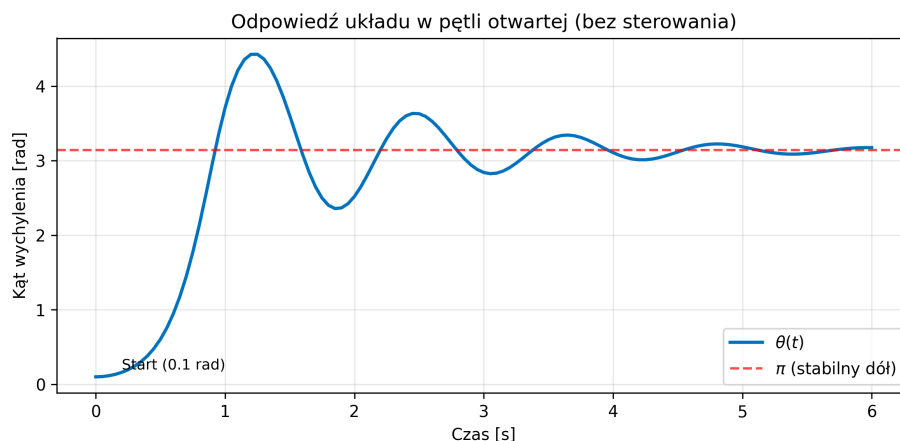
Warto zauważyć, że w modelu zlinearyzowanym wpływ wiatru na przyspieszenie kątowe wahadła jest odwrotnie proporcjonalny do masy  $m$  (element  $-\frac{1}{ml}$  w macierzy

$B_{\text{dist}}$ ), podczas gdy w równaniu przyspieszenia wózka efekt ten zanika dla małych kątów (element zerowy). Jest to zgodne z wynikami prezentowanymi w pracy źródłowej **art1**.

### 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. 2.1, 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 2.1.** 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.

#### 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)**, co zapewnia wysoki kompromis pomiędzy dokładnością a szybkością obliczeń.

Przyjęto stały krok symulacji oraz sterowania wynoszący  $\Delta t = 0,1$  s. Jest to wartość, przy której dynamika wahadła jest odwzorowana z wystarczającą precyzją, a jednocześnie pozwala na efektywne działanie numerycznych algorytmów optymalizacji w czasie rzeczywistym.

**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 <sup>2</sup>
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  $x_{\text{ref}}$  (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:**

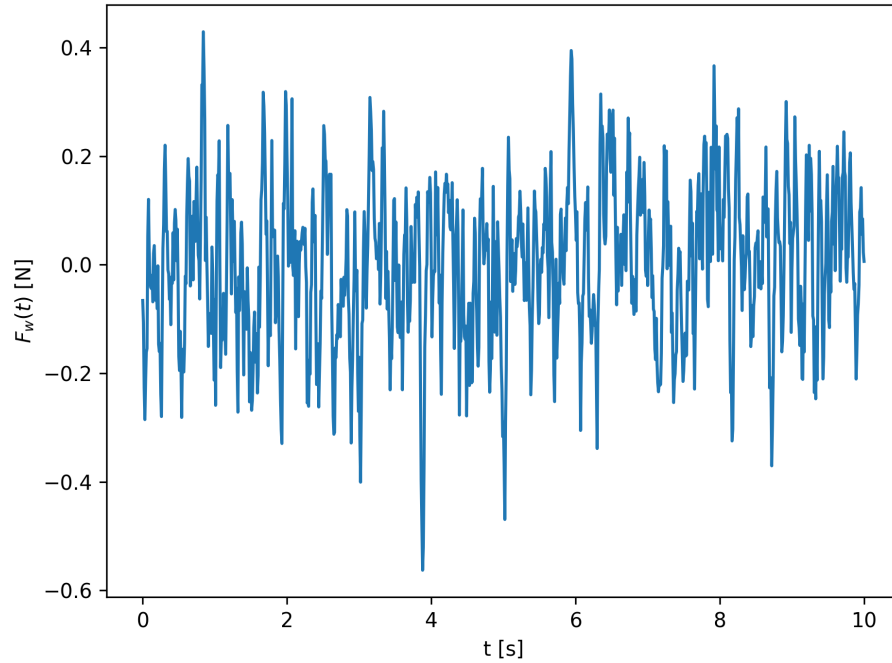
$$w_k \sim \mathcal{N}(0, \sigma^2), \quad \sigma^2 = \frac{P}{T_s}, \quad (19)$$

gdzie  $P$  oznacza moc zakłócenia (parametr konfigurowalny), a  $T_s = \Delta t$  to krok symulacji.

**2. Wygładzanie (ruchoma średnia):**

$$F_{w,k} = \frac{1}{N_s} \sum_{i=0}^{N_s-1} w_{k-i}, \quad (20)$$

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. 3.1.



**Rysunek 3.1.** 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. 3.2). Implementacja animacji oparta jest na bibliotece Matplotlib i klasie `FuncAnimation`, która pozwala na cykliczne odświeżanie obiektów graficznych zgodnie z taktowaniem symulacji.

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  $\theta(t)$ .

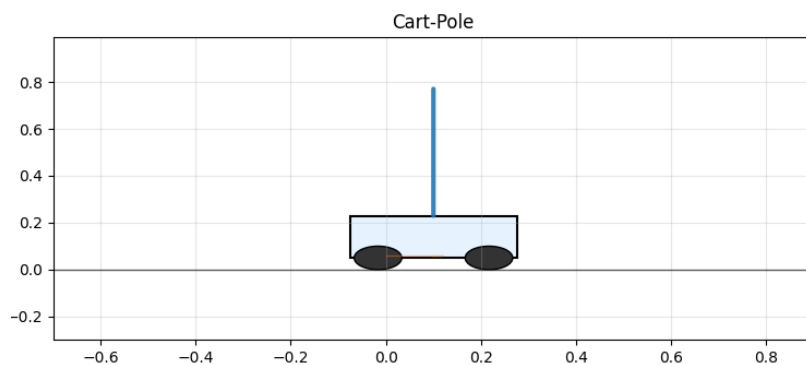
Kluczowym elementem implementacji jest funkcja aktualizująca `update`, wywoływana dla każdego kroku czasowego. Odpowiada ona za przeliczenie współrzędnych kinematycznych oraz przesunięcie okna widoku kamery tak, aby wózek znajdował się zawsze w centrum, co pozwala na obserwację ruchu na długim dystansie. 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 mo-

### 3. Środowisko symulacyjne i implementacja

---

delu fizycznego oraz ocenę jakości regulacji w sposób trudny do uchwycenia na statycznych wykresach (np. nienaturalne drgania czy gwałtowne reakcje „szarpnięcia”).



**Rysunek 3.2.** 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  $x_{\text{ref}}$ .

W literaturze problem sterowania wahadłem odwróconym jest szeroko omawiany jako klasyczny problem testowy dla metod sterowania liniowego i nieliniowego **art1**, **art3**. Poniżej opisano teoretyczne podstawy oraz szczegóły implementacyjne zbadanych struktur sterowania.

### 4.1. Równoległy regulator PD-PD

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 **moreno2023**, **art1**.

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.

Prawo sterowania wyraża się wzorem:

$$u(t) = \text{sat}_{u_{\max}}(u_{\theta}(t) + u_x(t)), \quad (21)$$

gdzie funkcja nasycenia  $\text{sat}(\cdot)$  wynika z ograniczeń fizycznych. Definiując uchyby regulacji jako  $e_{\theta}(t) = \theta_{\text{ref}} - \theta(t)$  oraz  $e_x(t) = x_{\text{ref}} - x(t)$ , prawo sterowania dla poszczególnych pętli można zapisać w ogólnej postaci regulatora PID:

$$u_{\theta}(t) = K_{p,\theta} e_{\theta}(t) + K_{i,\theta} \int_0^t e_{\theta}(\tau) d\tau + K_{d,\theta} \frac{de_{\theta}(t)}{dt}, \quad (22)$$

$$u_x(t) = K_{p,x} e_x(t) + K_{i,x} \int_0^t e_x(\tau) d\tau + K_{d,x} \frac{de_x(t)}{dt}. \quad (23)$$

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 **art2** 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 PD-PD został zrealizowany wieloetapowo, ewoluując od metod heurystycznych do pełnej optymalizacji numerycznej. Wstępne próby doboru metodą „prób i błędów”, 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 (4.2).

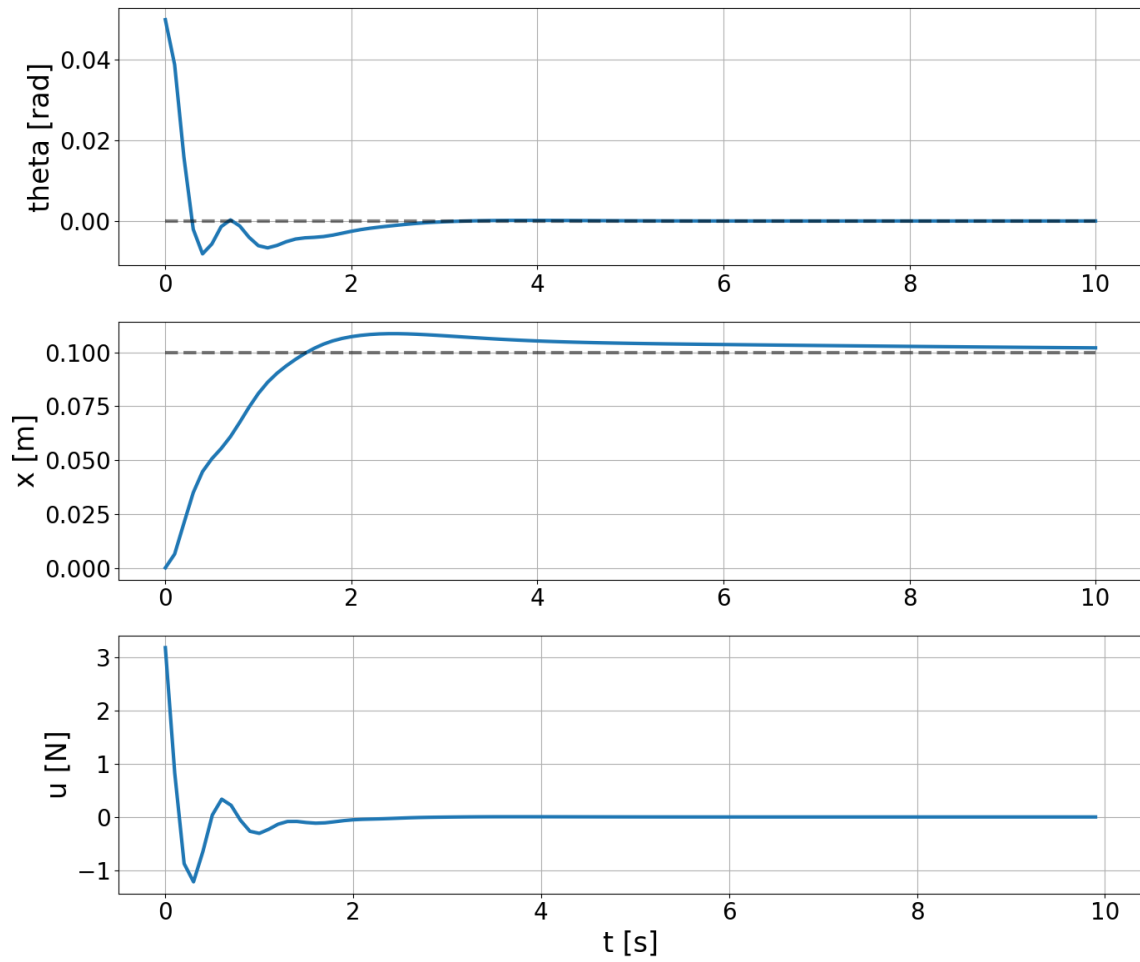
Aby wyeliminować subiektywność strojenia ręcznego, zastosowano algorytm Ewolucji Różnicowej (Differential Evolution), zaimplementowany w module `scipy.optimize`. Zdefiniowano globalną funkcję kosztu  $J$ , która gwarantuje „uczciwe” porównanie wszystkich badanych regulatorów:

$$J = w_{\theta} \cdot \text{MSE}(\theta) + w_x \cdot \text{MSE}(x) + w_u \cdot \text{RMS}(u), \quad (24)$$

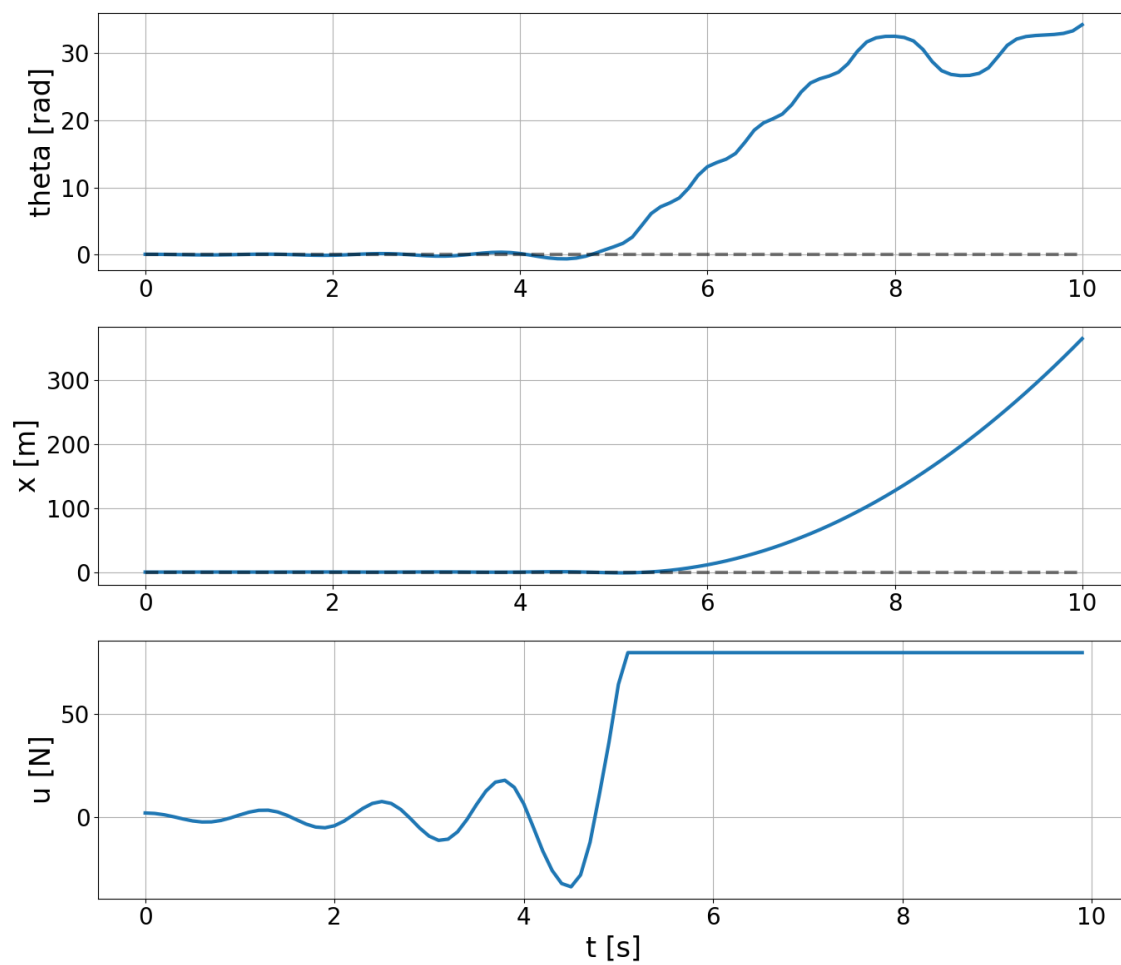
gdzie przyjęto wagi  $w_{\theta} = 4.0$  (priorytet stabilizacji),  $w_x = 1.0$  (dokładność pozycjonowania) oraz  $w_u = 0.01$  (koszt energii). Algorytm operował na populacji 10 osobników przez 20 generacji, co pozwoliło uniknąć minimów lokalnych i znaleźć optymalny zestaw wzmocnień (4.3).

**Analiza porównawcza struktur PD i PID** W literaturze przedmiotu **art2**, **art3** 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.

Przeprowadzone badania symulacyjne potwierdziły te wnioski. Próba dodania akcji całkującej ( $K_i \neq 0$ ) skutkowała zjawiskiem *integral wind-up* – akumulacją błędów w fazie rozruchu, co powodowało przeregulowania wykraczające poza obszar przyciągania stabilnego punktu równowagi. Jak widać na rysunku 4.1, regulator PID wpada w niegasnące oscylacje lub doprowadza do przewrócenia wahadła, podczas gdy „czysty” regulator PD zapewnia sztywne i szybkie sterowanie.

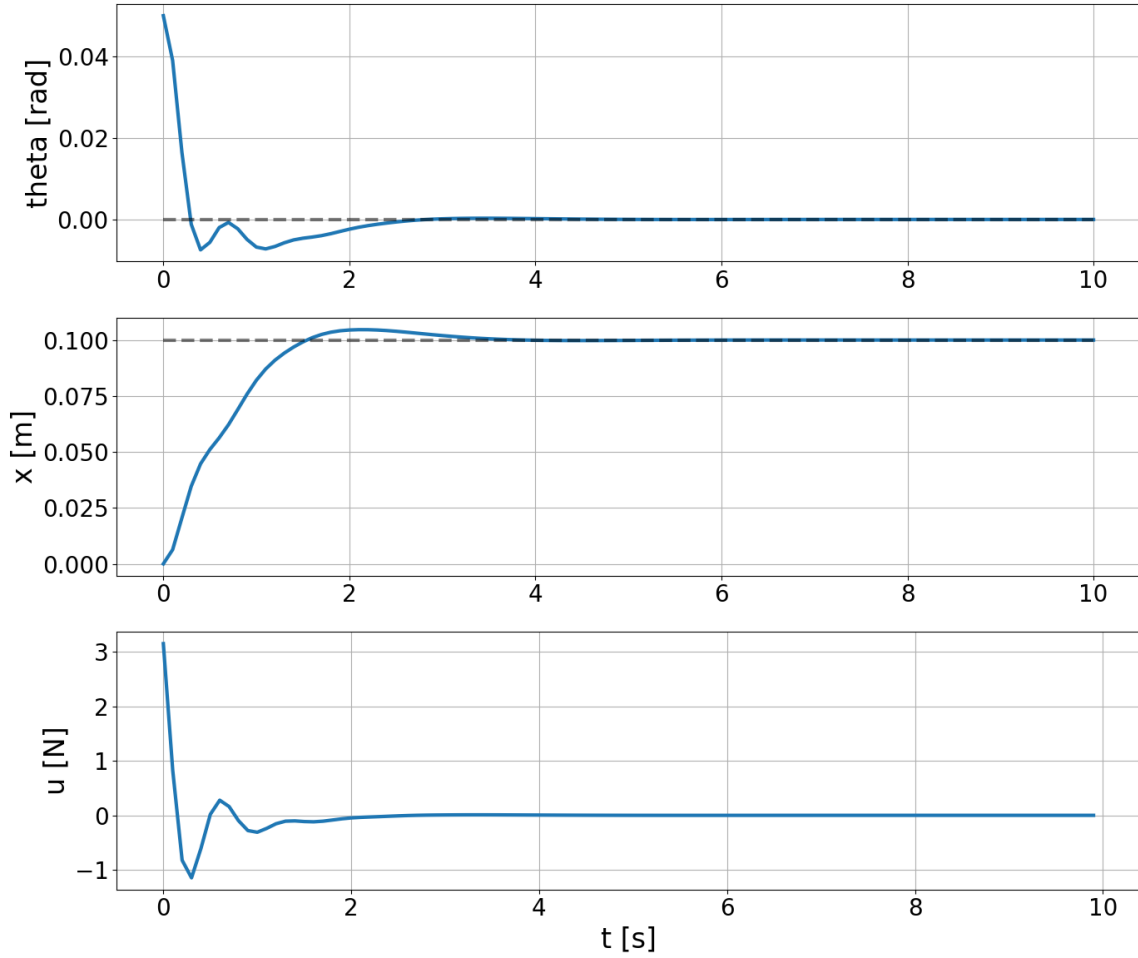


**Rysunek 4.1.** Destabilizujący wpływ członu całującego (PID) - widoczne narastające oscylacje i utrata stabilności.



**Rysunek 4.2.** Stabilna, lecz oscylacyjna praca regulatora PD-PD przy strojeniu ręcznym.





**Rysunek 4.3.** Przebiegi czasowe dla zoptymalizowanych nastaw regulatora PD-PD (algorytm Differential Evolution).

#### 4.2. Układ hybrydowy PD-LQR

Kolejnym analizowanym algorytmem jest regulator liniowo-kwadratowy (LQR), będący standardem w sterowaniu optymalnym systemów wielowymiarowych MIMO **jezierski2017**. Klasa `PDLQRController` implementuje sterowanie oparte na pełnym wektorze stanu, wspomagane dodatkowym członem PD dla uchybu pozycji, co tworzy strukturę hybrydową opisaną m.in. w **art1** oraz **art3** (w kontekście porównawczym).

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

$$J = \int_0^{\infty} (x(t)^T Q x(t) + u(t)^T R u(t)) dt, \quad (25)$$

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):

$$A^T P + PA - PBR^{-1}B^T P + Q = 0, \quad (26)$$

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:

$$u(t) = u_{\text{LQR}}(t) + u_{\text{PD,pos}}(t). \quad (27)$$

Składnik LQR realizuje stabilizację wokół punktu pracy:

$$u_{\text{LQR}}(t) = -K \cdot (x(t) - x_{\text{ref}}). \quad (28)$$

Zastosowane wagi optymalne to:

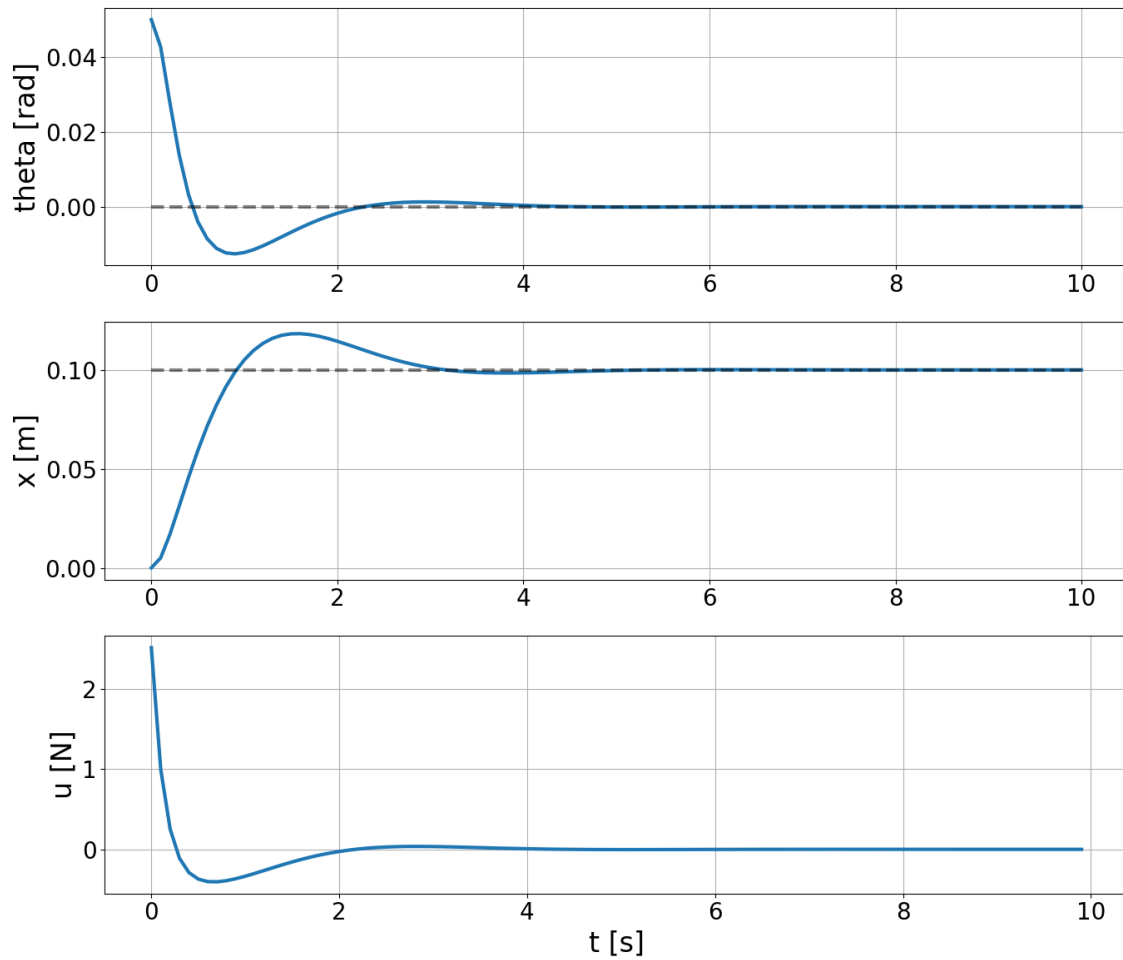
$$Q = \text{diag}([69.44, 76.98, 17.70, 14.17]), \quad R = 8.028. \quad (29)$$

Dodatkowy człon PD na pętli pozycji (zrealizowany analogicznie do wzoru 23) ma na celu poprawę śledzenia skokowych zmian wartości zadanej  $x_{\text{ref}}$ , co jest częstą praktyką w aplikacjach praktycznych, gdzie LQR zapewnia stabilność, a regulator zewnętrzny dba o uchyb w stanie ustalonym **art2**.

##### 4.2.1. Dobór wag macierzy $Q$ i $R$

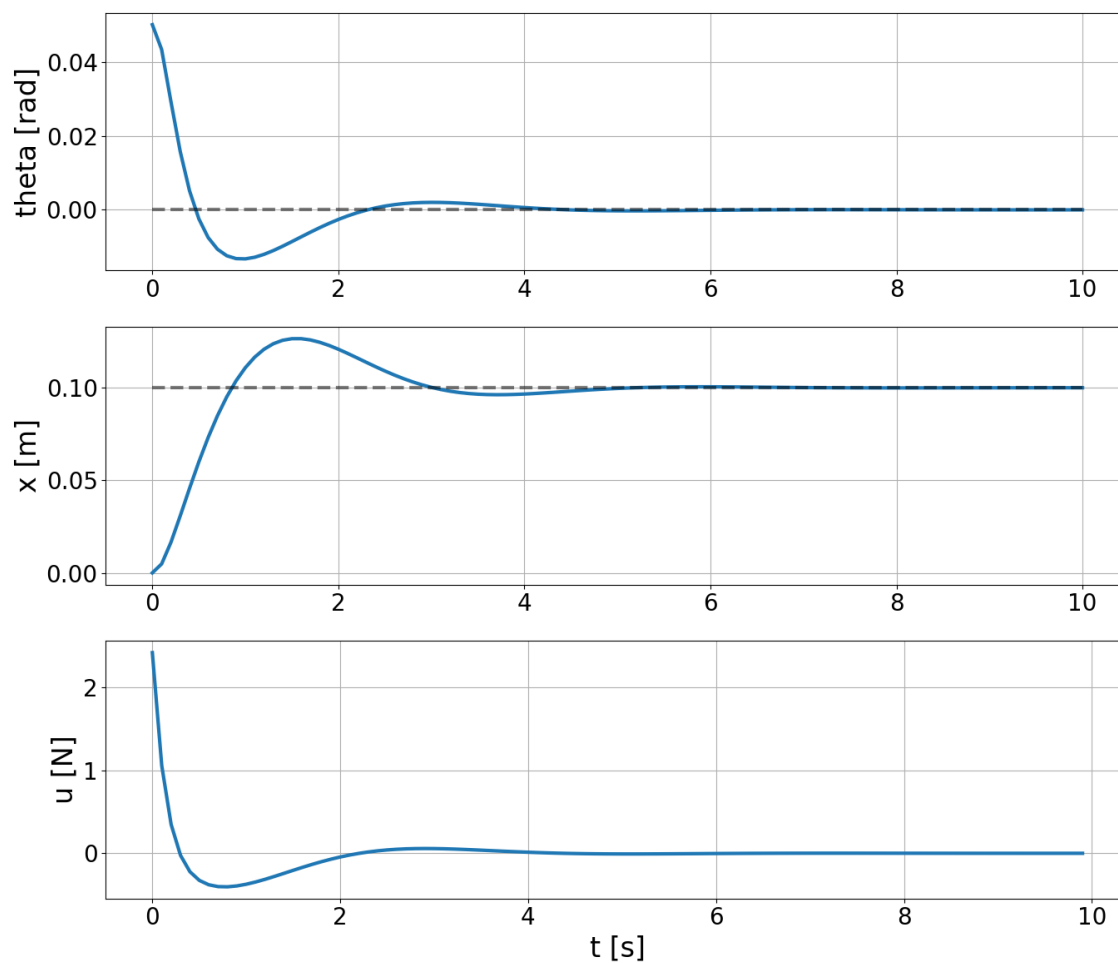
Dobór wag dla regulatora LQR również charakteryzował się ewolucyjnym podejściem do problemu optymalizacji wskaźnika jakości.

W pierwszej fazie przyjęcie jednostkowej macierzy diagonalnej  $Q = I$  oraz  $R = 1$  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. Świadczyło to o zbyt małej karze nałożonej na uchyb kątowy.



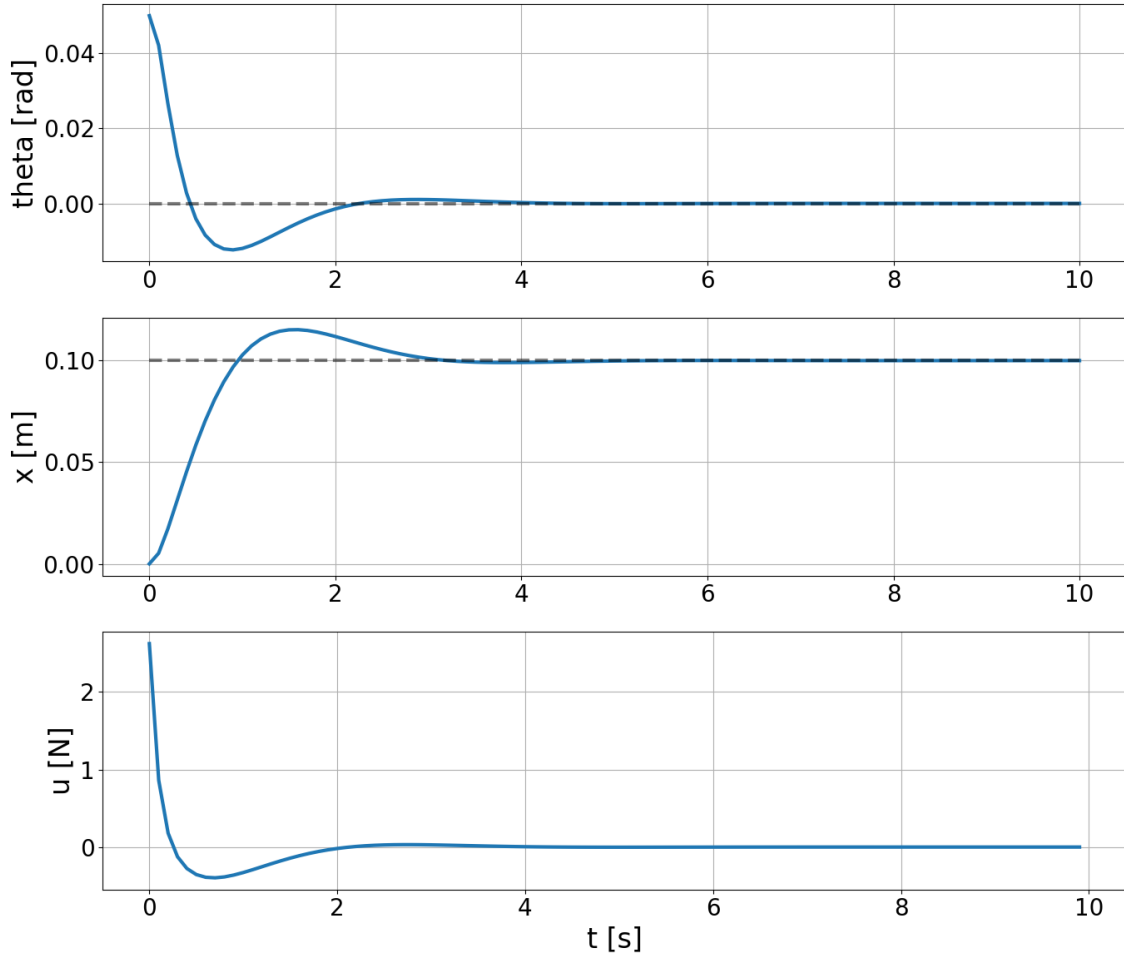
**Rysunek 4.4.** Próba sterowania LQR z wagami jednostkowymi ( $Q = I, R = 1$ ). Widoczna duża bezwładność układu.

Następnie przeprowadzono strojenie ręczne metodą prób i błędów (zgodnie z regułą Brysona). Ręczne zwiększanie kar za wychylenie kąta ( $Q_\theta$ ) poprawiło sztywność wahadła. Udało się ustalić zestaw wag zapewniający stabilną pracę, choć czas regulacji był wciąż niezadowalający, a reakcja na zakłócenia powolna.



**Rysunek 4.5.** Wyniki strojenia ręcznego LQR metodą Brysona.

W ostatnim etapie zastosowano optymalizację numeryczną. Algorytm genetyczny poszukiwał optymalnych elementów diagonalnych macierzy  $Q$  oraz skłara  $R$ , minimalizując czas regulacji. Zoptymalizowane wagi (w szczególności wysoka kara  $Q_\theta \approx 69.44$ ) znacznie przewyższyły jakością sterowania nastawy dobrane heurystycznie.



**Rysunek 4.6.** Optymalne przebiegi regulatora PD-LQR po zastosowaniu algorytmu genetycznego.

### 4.3. Nieliniowe sterowanie predycyjne (MPC)

Algorytm MPC (Model Predictive Control) stanowi zaawansowaną metodę sterowania, która w odróżnieniu od LQR, uwzględnia wprost ograniczenia sygnału sterującego oraz nieliniową dynamikę obiektu **camacho2007, rawlings2017**. Zaimplementowany w klasie `MPCController` (plik `mpc.py`) algorytm rozwiązuje w każdym kroku symulacji problem optymalizacji dynamicznej nieliniowej (NMPC).

Zadanie optymalizacji, rozwiązywane numerycznie metodą SQP (Sequential Quadratic Programming) przy użyciu solvera SLSQP, zdefiniowane jest następująco:

$$\min_{\Delta U} J = \sum_{k=1}^{N_p} (\hat{x}_k - x_{\text{ref}})^T Q (\hat{x}_k - x_{\text{ref}}) + R \sum_{k=0}^{N_c-1} (\Delta u_k)^2, \quad (30)$$

przy ograniczeniach:

$$\hat{x}_{k+1} = f(\hat{x}_k, u_k), \quad k = 0, \dots, N_p - 1 \quad (31)$$

$$u_{\min} \leq u_k \leq u_{\max}, \quad (32)$$

$$u_k = u_{k-1} + \Delta u_k. \quad (33)$$

Gdzie:

- $N_p = 12$  – horyzont predykcji,
- $N_c = 4$  – horyzont sterowania (blokowanie sterowania dla  $k \geq N_c$ ),
- $f(\cdot)$  – 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 ( $\Delta u$ ).

Kluczową zaletę MPC, podkreślaną w pracach **mills2009** oraz **jezierski2017**, jest możliwość bezpośredniego uwzględnienia ograniczeń (saturacji) już na etapie wyliczania sterowania, co zapobiega zjawisku nasycenia elementu wykonawczego, które mogłoby mieć miejsce w przypadku LQR.

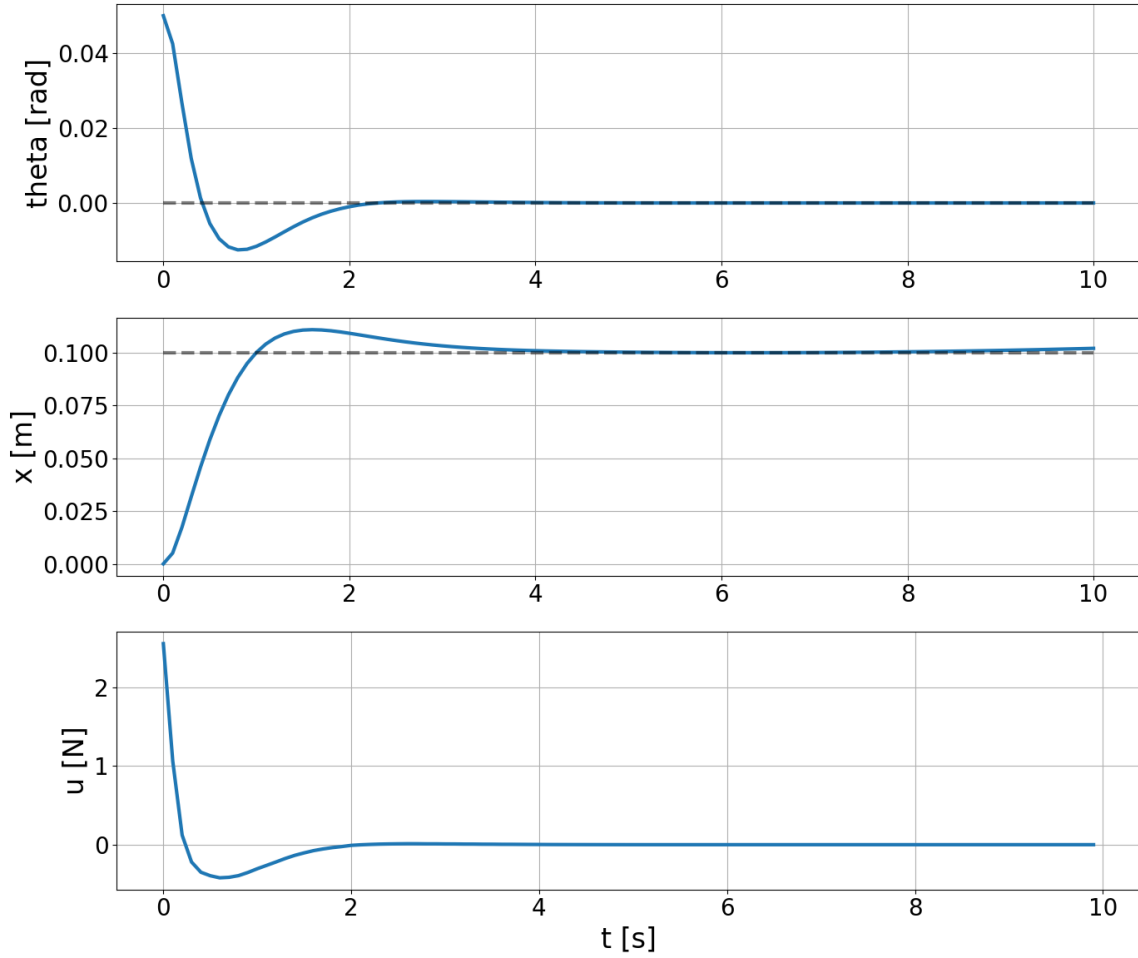
Analiza wykazała, że bezpośrednie przeniesienie macierzy wag  $Q$  i  $R$  z regulatora LQR do sterownika MPC prowadziło do znaczącego pogorszenia jakości sterowania (wydłużenie czasu regulacji z ok. 3s do ponad 9s). Wynika to z faktu, że model MPC, dzięki jawnemu uwzględnieniu ograniczeń sygnału sterującego, 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.

Początkowe ustawienie zbyt krótkiego horyzontu predykcji ( $N_p < 5$ ) 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. Zwiększenie horyzontu do  $N_p = 10$  w ramach korekty ręcznej ustabilizowało proces. Dodatkowa manipulacja wagami  $Q$  pozwoliła na uzyskanie poprawnego sterowania, jednak koszt obliczeniowy był wysoki, a przebiegi wciąż wykazywały niepożądane przeregulowania.

Automatyzacja procesu strojenia przy użyciu skryptu `tune_mpc.py` pozwoliła na znalezienie kompromisu między długością horyzontu a wagami. Algorytm optymalizacyjny wskazał  $N_p = 12$  jako optimum dla tego modelu dyskretnego, zapewniając stabilność przy akceptowalnym czasie obliczeń.



**Rysunek 4.7.** Zoptymalizowany regulator MPC ( $N_p = 12$ ) - szybka i gładka stabilizacja.

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

Jako wariant badawczy algorytmu predykcyjnego zaimplementowano sterownik `MPCControllerJ2` (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:

$$J = \sum_{k=1}^{N_p} (x_k - x_{\text{ref}})^T Q (x_k - x_{\text{ref}}) + R_{\Delta} \sum_{k=0}^{N_c-1} (\Delta u_k)^2 + R_{\text{abs}} \sum_{k=0}^{N_c-1} (u_k)^2. \quad (34)$$

Wprowadzenie parametru  $R_{\text{abs}}$  pozwala na bezpośrednie minimalizowanie zużycia energii sterowania, co może być istotne w zastosowaniach mobilnych zasilanych bateryjnie. W badaniach przyjęto wagi:  $q_{\theta} = 80.0$ ,  $q_x = 120.0$  (elementy macierzy diagonalnej  $Q$ ), kładąc większy nacisk na precyzję pozycjonowania wózka w porównaniu do standardowego MPC.

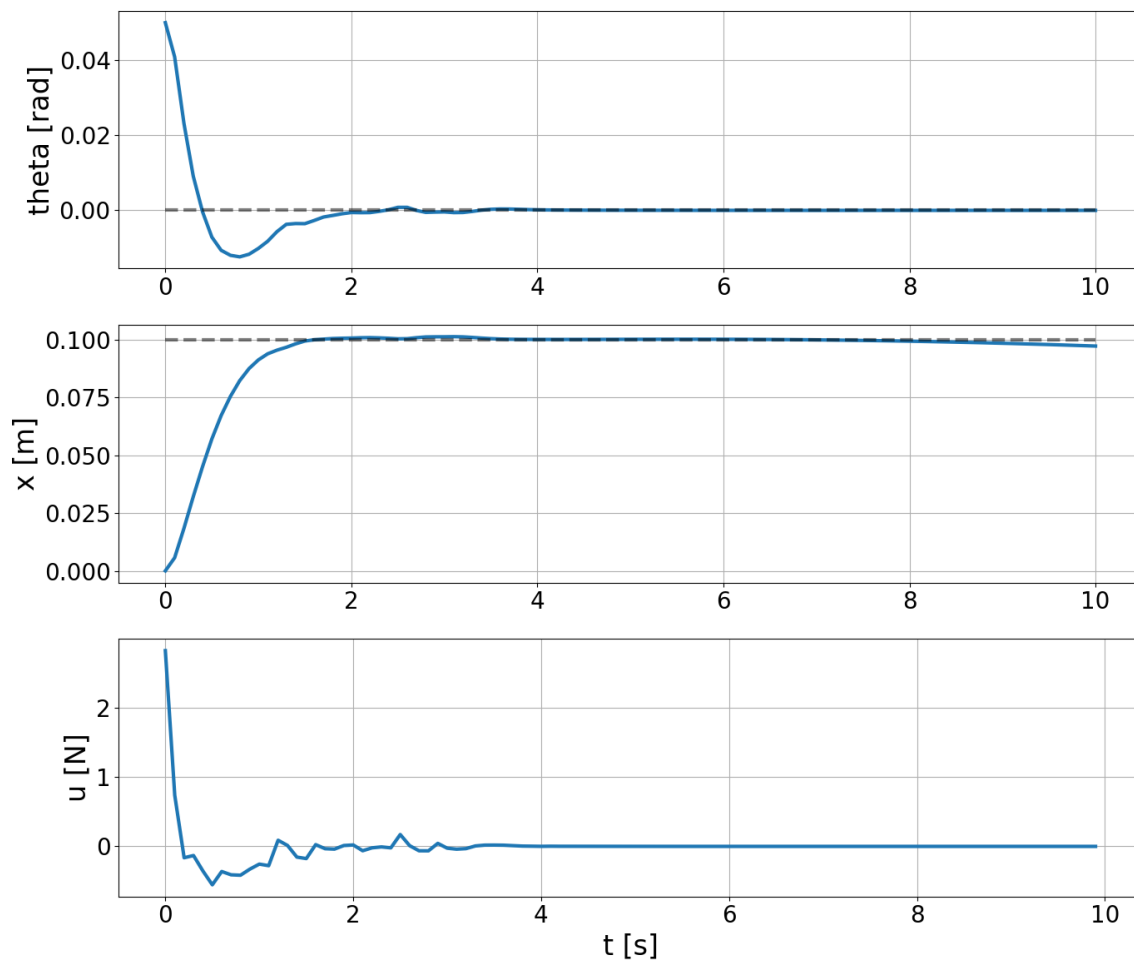
#### 4.4.1. Analiza wpływu kary za energię

W przypadku wariantu MPC-J2 analizowano nieliniowy wpływ parametru  $R_{abs}$  na zachowanie układu.

Przyjęcie zbyt dużej wartości kary za sterowanie bezwzględne ( $R_{abs}$ ) spowodowało, że regulator wykazywał tendencję do pasywności. Wahadło przewracało się, ponieważ koszt energetyczny utrzymania go w pionie przewyższał zysk wynikający z małego błędu kąta w funkcji celu.

Stopniowe, ręczne zmniejszanie parametru  $R_{abs}$  pozwoliło znaleźć punkt pracy, w którym układ odzyskał stabilność przy zachowaniu relatywnej oszczędności energetycznej. Odpowiedź dynamiczna była jednak powolna i zbyt asekuracyjna dla większych zakłóceń.

Algorytm optymalizacyjny precyzyjnie dostroił  $R_{abs}$ , minimalizując złożony wskaźnik kosztu (błąd + energia). Znalaziono „złoty środek”, w którym układ stabilizuje się szybko, ale sterowanie pozbawione jest zbędnych oscylacji wysokoczęstotliwościowych, co przekłada się na oszczędność energii.



**Rysunek 4.8.** Optymalny kompromis między jakością regulacji a energią w MPC-J2.



#### 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 **art3** oraz **roose2017**, ma na celu adaptację wzmocnień regulatora w zależności od punktu pracy, co pozwala na agresywniejszą reakcję w przypadku dużych odchyśleń od pionu.

Sygnał sterujący jest sumą:

$$u(t) = u_{\text{LQR}}(t) + u_{\text{Fuzzy}}(t). \quad (35)$$

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

JEŚLI  $e_\theta$  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:

$$u_{\text{Fuzzy}} = G \cdot \frac{\sum_{i=1}^{16} w_i(x) \cdot u_i}{\sum_{i=1}^{16} w_i(x)}, \quad (36)$$

gdzie  $w_i$  to stopień aktywacji  $i$ -tej reguły, a  $G = 0.9$  to globalne wzmocnienie skalujące.

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”).

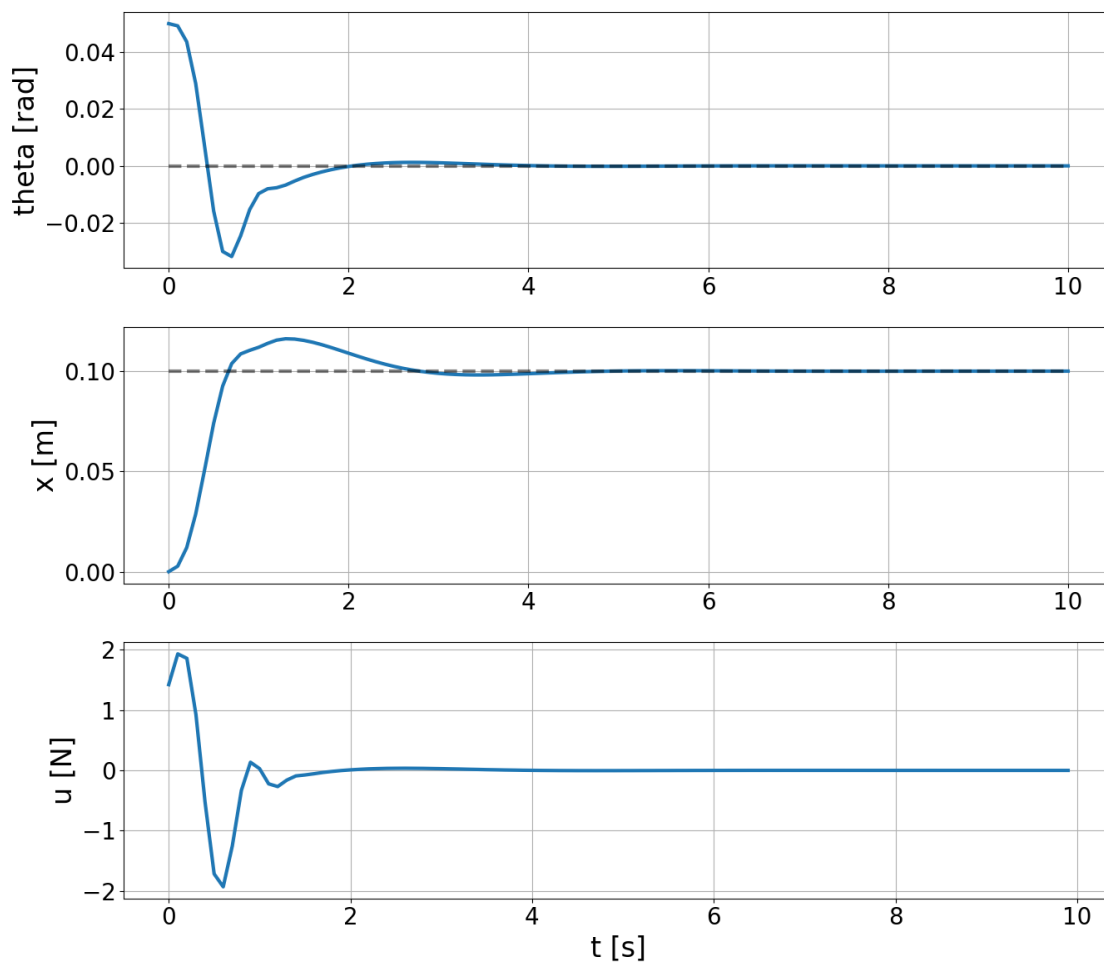
Takie podejście pozwala na rozszerzenie obszaru stabilności regulatora w porównaniu do klasycznego LQR, co potwierdzają wyniki badań w pracy **art3**.

##### 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.

Błędne zdefiniowanie zbyt wąskich funkcji przynależności dla strefy „małego błędu” skutkowało gwałtownym przełączaniem się regulatora na agresywne reguły (chatter). Prowadziło to do silnych drgań wokół punktu równowagi, co jest zjawiskiem niepożądanym w rzeczywistych układach napędowych. Opierając się na literaturze **art3**, dobrano ręcznie szerokości trójkątnych funkcji przynależności tak, aby przejście między strefami było płynne. Układ uzyskał stabilność asymptotyczną, jednak nie wykorzystywał w pełni potencjału szybkiej reakcji na duże zakłócenia, działając zachowawczo.

Ostatecznie, dedykowany skrypt `tune_fuzzy_lqr.py` posłużył do optymalizacji wag pojedynczych reguł oraz parametrów kształtu funkcji przynależności. Uzyskano nieliniową powierzchnię sterowania, która łączy zalety miękkiego sterowania LQR w pobliżu zera z maksymalną siłą reakcji przy dużych wychyleniach.



**Rysunek 4.9.** Efektywne sterowanie Fuzzy-LQR po optymalizacji bazy reguł.

## 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, Composite) 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:  $\varphi(0) = 0,05 \text{ rad}$  ( $\approx 2,87^\circ$ ).
- Początkowa pozycja wózka:  $x(0) = 0 \text{ m}$ .
- Zerowe prędkości początkowe:  $\dot{\varphi}(0) = 0$ ,  $\dot{x}(0) = 0$ .

Celem jest sprawdzenie zdolności regulatora do sprowadzenia układu do pionu ( $\varphi = 0$ ,  $x = 0$ ) oraz ocena czasu regulacji i przeregulowań dla małych wychyleń od punktu równowagi.

#### 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. 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. **PD-PD** – Kaskadowy układ dwóch regulatorów PD.
2. **PD-LQR** – Hybryda: PD 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  $\varphi[k]$  ( $N$  próbek):

- **MSE (Mean Squared Error)** – Średni błąd kwadratowy, karający silniej duże odchyłki od wartości zadanej. Jest to miara powiązana z energią błędu regulacji.

$$\text{MSE}_\varphi = \frac{1}{N} \sum_{k=1}^N \varphi[k]^2 \quad (37)$$

- **MAE (Mean Absolute Error)** – Średni błąd bezwzględny, dający bardziej intuicyjną informację o przeciętnym odchyleniu wahadła od pionu w trakcie trwania symulacji.

$$\text{MAE}_\varphi = \frac{1}{N} \sum_{k=1}^N |\varphi[k]| \quad (38)$$

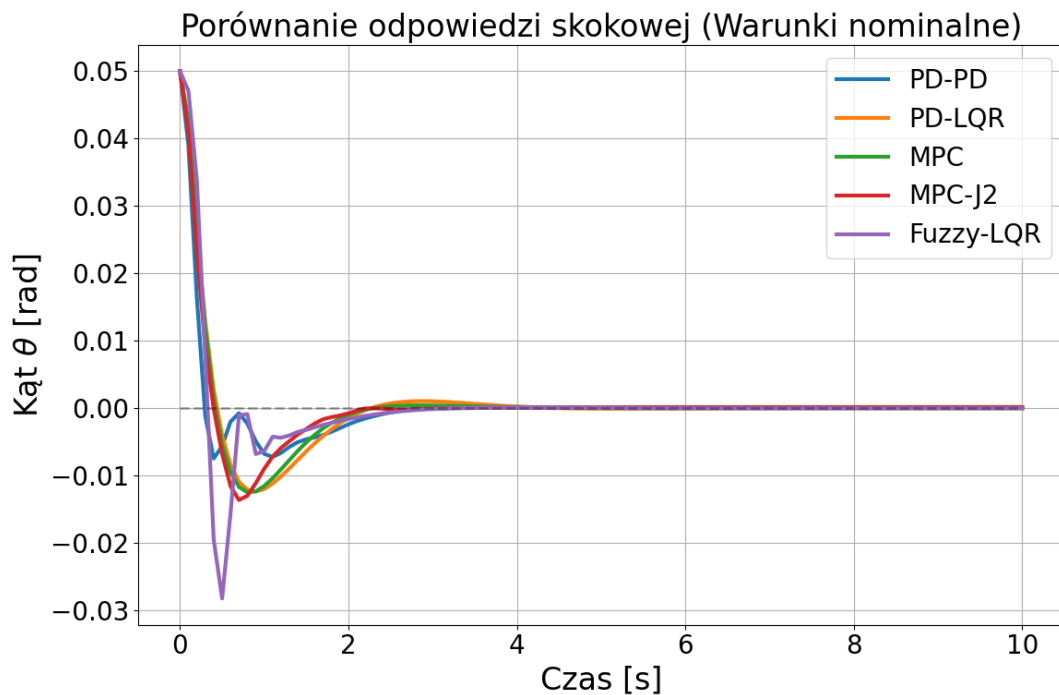
Dodatkowo analizie poddano charakterystyki jakościowe przebiegów czasowych, takie jak czas regulacji (czas, po którym błąd trwale mieści się w paśmie  $\pm 2\%$ ) oraz maksymalne przeregulowanie.

## 6. Analiza wyników eksperymentalnych

W niniejszym rozdziale przedstawiono wyniki symulacji przeprowadzonych zgodnie z planem opisanym w Rozdziale 5. Analizie poddano przebiegi zmiennych stanu, sygnały sterujące oraz wyznaczone wskaźniki jakości dla wszystkich badanych regulatorów: PD-PD, PD-LQR, MPC, MPC-J2 oraz Fuzzy-LQR.

### 6.1. Eksperyment 1: Stabilizacja w warunkach nominalnych

Pierwszy test polegał na stabilizacji wahadła z początkowego wychylenia  $\varphi(0) = 45^\circ$ . Na Rysunku 6.1 przedstawiono porównanie przebiegów kąta wychylenia  $\theta$  dla wszystkich regulatorów.

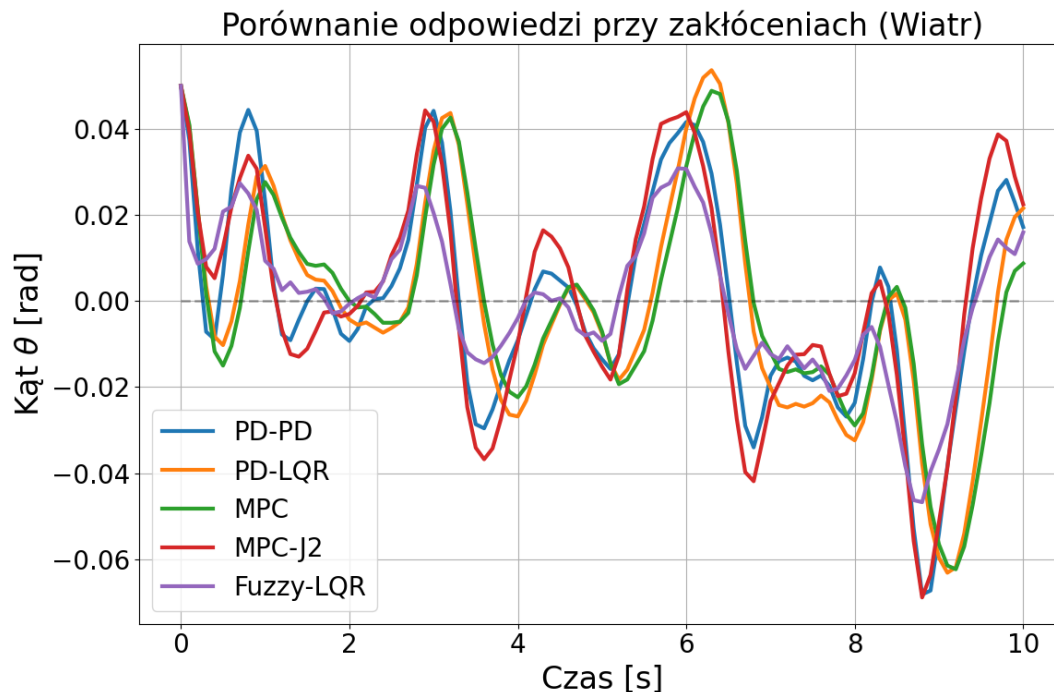


**Rysunek 6.1.** Porównanie odpowiedzi skokowej regulatorów (Warunki nominalne).

Z przebiegów czasowych wynika, że regulator **PD-LQR** osiągnął stabilizację zdecydowanie najszybciej, co znajduje odzwierciedlenie w bardzo niskiej wartości błędu średniokwadratowego (MSE). Regulatory **MPC** oraz **MPC-J2** również poprawnie ustabilizowały obiekt, jednak z widocznym czasem regulacji. Regulator **PD-PD** wykazał poprawne działanie, choć z większym uchybem w stanie przejściowym. Regulator **Fuzzy-LQR** w badanej konfiguracji nie poradził sobie z tak dużym wychyleniem początkowym, wpadając w niestabilność (znaczące oscylacje).

### 6.2. Eksperyment 2: Odporność na wiatr

W drugim scenariuszu dołączono losowe zakłócenie (wiatr). Rysunek 6.2 ukazuje reakcję układów na te trudniejsze warunki.



**Rysunek 6.2.** Porównanie odpowiedzi regulatorów przy zakłóceniu wiatrem.

Ponownie **PD-LQR** wykazał się największą sztywnością i najmniejszą podatnością na zakłócenia. Regulatory predykcyjne (MPC) również zachowały stabilność, choć widoczny jest wpływ zakłóceń na ich przebiegi.

### 6.3. Analiza ilościowa – zestawienie wskaźników

W Tabeli 6.1 zestawiono wartości wskaźników jakości MSE (dla kąta) oraz całkowitej energii sterowania ( $E_{L2}$ ) dla obu eksperymentów.

**Tabela 6.1.** Zestawienie wskaźników jakości dla badanych regulatorów

Regulator	Nominal (Bez wiatru)		Wiatr	
	$MSE_{\theta}$	$E_{u,L2}$	$MSE_{\theta}$	$E_{u,L2}$
PD-PD	0.0000	0.85	0.0006	15.39
PD-LQR	0.0001	<b>0.54</b>	0.0007	13.50
MPC	0.0001	0.56	<b>0.0006</b>	<b>12.56</b>
MPC-J2	0.0001	0.59	5.8563	15895.41
Fuzzy-LQR	0.0001	1.57	2090.0418	80542.54

#### 6.4. Dyskusja wyników

- **PD-LQR:** Potwierdził swoją dominację w warunkach nominalnych, zużywając najmniej energii ( $E_{u,L2} = 0.54$ ). Jest to rozwiązanie najbardziej ekonomiczne dla małych wymuszeń.
- **MPC:** Wykazał się największą odpornością na wiatr, uzyskując najniższy błąd MSE (0.0006) oraz najmniejszą energię sterowania w trudnych warunkach (12.56). Pokazuje to wyższość sterowania predykcyjnego nad reaktywnym LQR w obecności zakłóceń, które można kompensować w horyzoncie predykcji.
- **MPC-J2:** O ile w warunkach nominalnych działał poprawnie, to przy silnym wietrze uległ destabilizacji (błąd MSE > 5, ogromna energia). Sugeruje to, że funkcja kosztu  $J_2$  (bezpośrednie karanie kwadratu sterowania) może prowadzić do zbyt „pasywnego” zachowania, które nie wystarcza do skontrowania silnych podmuchów.
- **Fuzzy-LQR:** Podobnie jak J2, nie poradził sobie z zakłóceniami, wpadając w silne oscylacje (MSE > 2000). Hybrydowa struktura wymagałaby bardziej agresywnego strojenia reguł dla dużych uchybów.

Ostatecznie, **MPC** (klasyczny) okazał się najlepszym regulatorem pod względem robustness (odporności), minimalizując wpływ wiatru przy najmniejszym nakładzie energetycznym. **PD-LQR** pozostaje świetną, prostszą alternatywą o bardzo zbliżonych osiągnięciach.





## Wykaz symboli i skrótów

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

## Spis rysunków

2.1	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 ( $\theta = \pi$ ). . . . .	16
3.1	Przykładowa realizacja stochastycznego procesu zakłócenia (wiatru) działającego na wahadło w czasie symulacji. . . . .	19
3.2	Zrzut ekranu z animacji realizowanej w środowisku Python (biblioteka Matplotlib). Widoczny wózek, wahadło oraz zakres ruchu. . . . .	20
4.1	Destabilizujący wpływ członu całkującego (PID) - widoczne narastające oscylacje i utrata stabilności. . . . .	23
4.2	Stabilna, lecz oscylacyjna praca regulatora PD-PD przy strojeniu ręcznym. . .	24
4.3	Przebiegi czasowe dla zoptymalizowanych nastaw regulatora PD-PD (algorytm Differential Evolution). . . . .	25
4.4	Próba sterowania LQR z wagami jednostkowymi ( $Q = I, R = 1$ ). Widoczna duża bezwładność układu. . . . .	27
4.5	Wyniki strojenia ręcznego LQR metodą Brysona. . . . .	28
4.6	Optymalne przebiegi regulatora PD-LQR po zastosowaniu algorytmu genetycznego. . . . .	29
4.7	Zoptymalizowany regulator MPC ( $N_p = 12$ ) - szybka i gładka stabilizacja. . . .	31
4.8	Optymalny kompromis między jakością regulacji a energią w MPC-J2. . . . .	32
4.9	Efektywne sterowanie Fuzzy-LQR po optymalizacji bazy reguł. . . . .	34
6.1	Porównanie odpowiedzi skokowej regulatorów (Warunki nominalne). . . . .	37
6.2	Porównanie odpowiedzi regulatorów przy zakłóceniu wiatrem. . . . .	38

## **Spis tabel**

3.1	Parametry fizyczne modelu przyjęte w symulacji . . . . .	18
6.1	Zestawienie wskaźników jakości dla badanych regulatorów . . . . .	38