

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

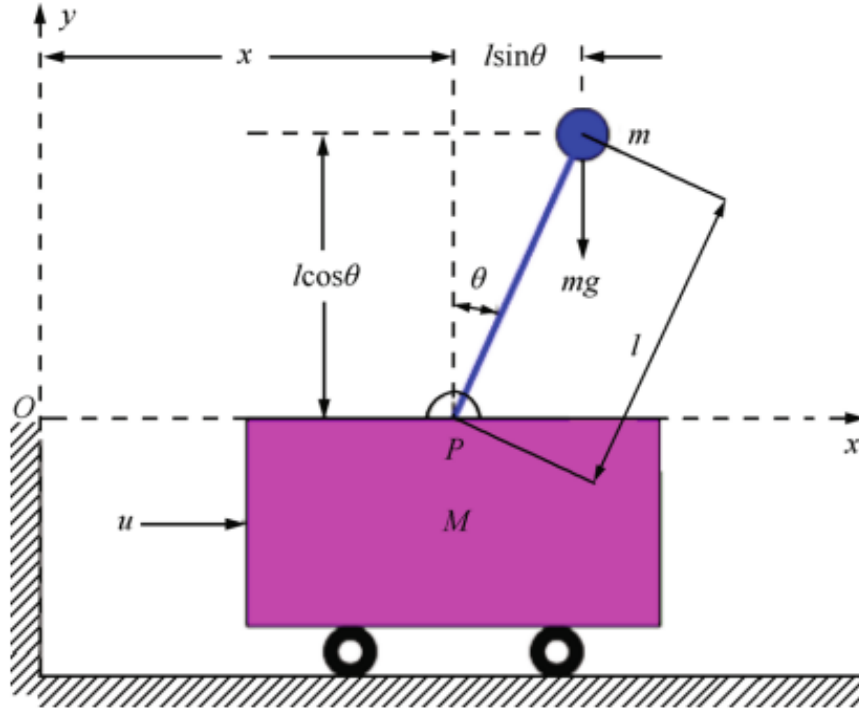
Rozważany obiekt sterowania należy do klasy mechanicznych układów podwzbudzonych (ang. *underactuated systems*), co oznacza, że liczba wejść sterujących jest mniejsza od liczby stopni swobody. Układ składa się z wózka poruszającego się wzdłuż poziomej osi oraz pręta z masą skupioną, zamocowanego przegubowo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

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:

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

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

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

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):

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

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]:

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

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

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 [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 \mathbf{B} oraz macierz zakłóceń \mathbf{B}_{dist} przyjmują postać:

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

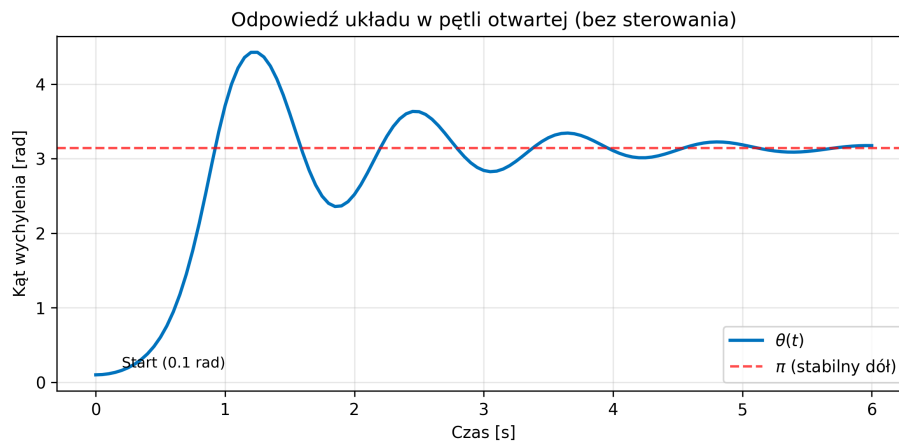
Warto zauważyć, że w modelu zlinearyzowanym wpływ wiatru na przyspieszenie kątowe wahadła jest odwrotnie proporcjonalny do masy m (element $-\frac{1}{ml}$ w macierzy \mathbf{B}_{dist}), podczas gdy w równaniu przyspieszenia wózka efekt ten zanika dla małych kątów (element zerowy). Jest to zgodne z wynikami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

2.6. 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. 2.2, 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 ener-

gia mechaniczna jest rozpraszana. W rezultacie, kąt wychylenia $\theta(t)$ dąży do stabilnego punktu równowagi dolnej, tj. $\theta = \pi$ rad.



Rysunek 2.2. 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 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:

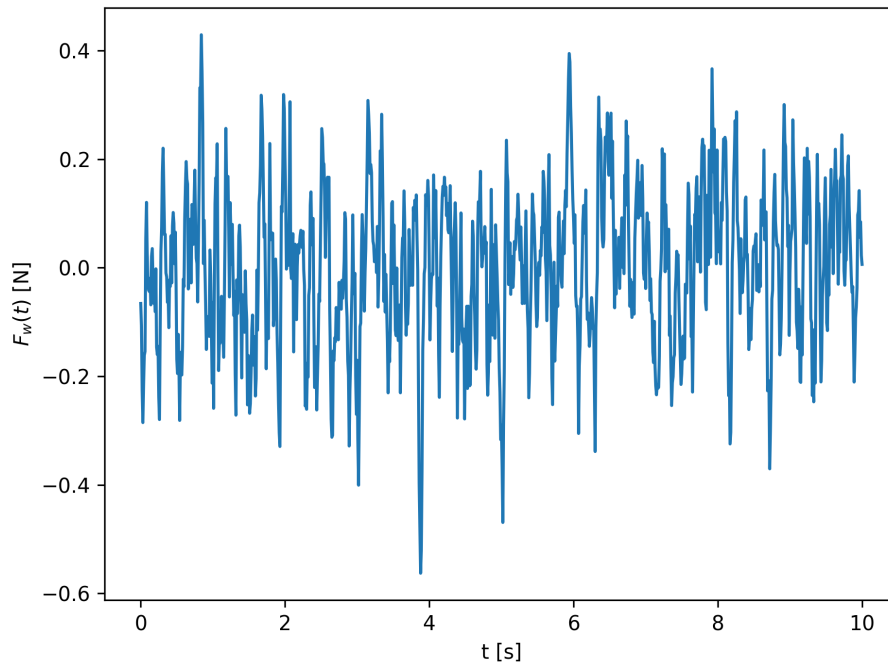
$$w_k \sim \mathcal{N}(0, \sigma^2), \quad \sigma^2 = \frac{P}{T_s}, \quad (19)$$

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.

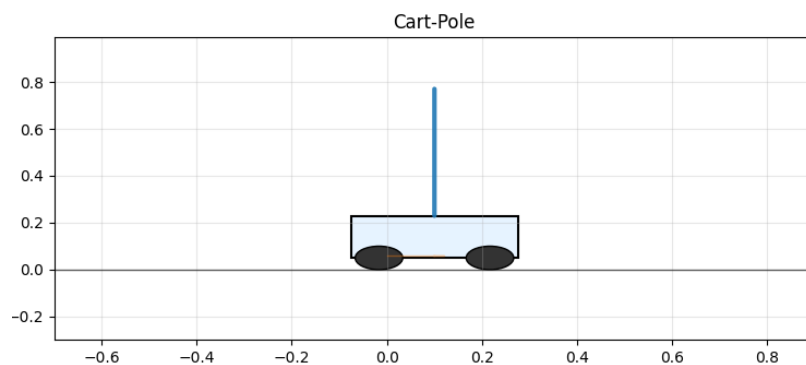
Graficzna reprezentacja obiektu (robot) zbudowana jest z prostych prymitywów geometrycznych:

- **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-

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_{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 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 [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.

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 [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 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).

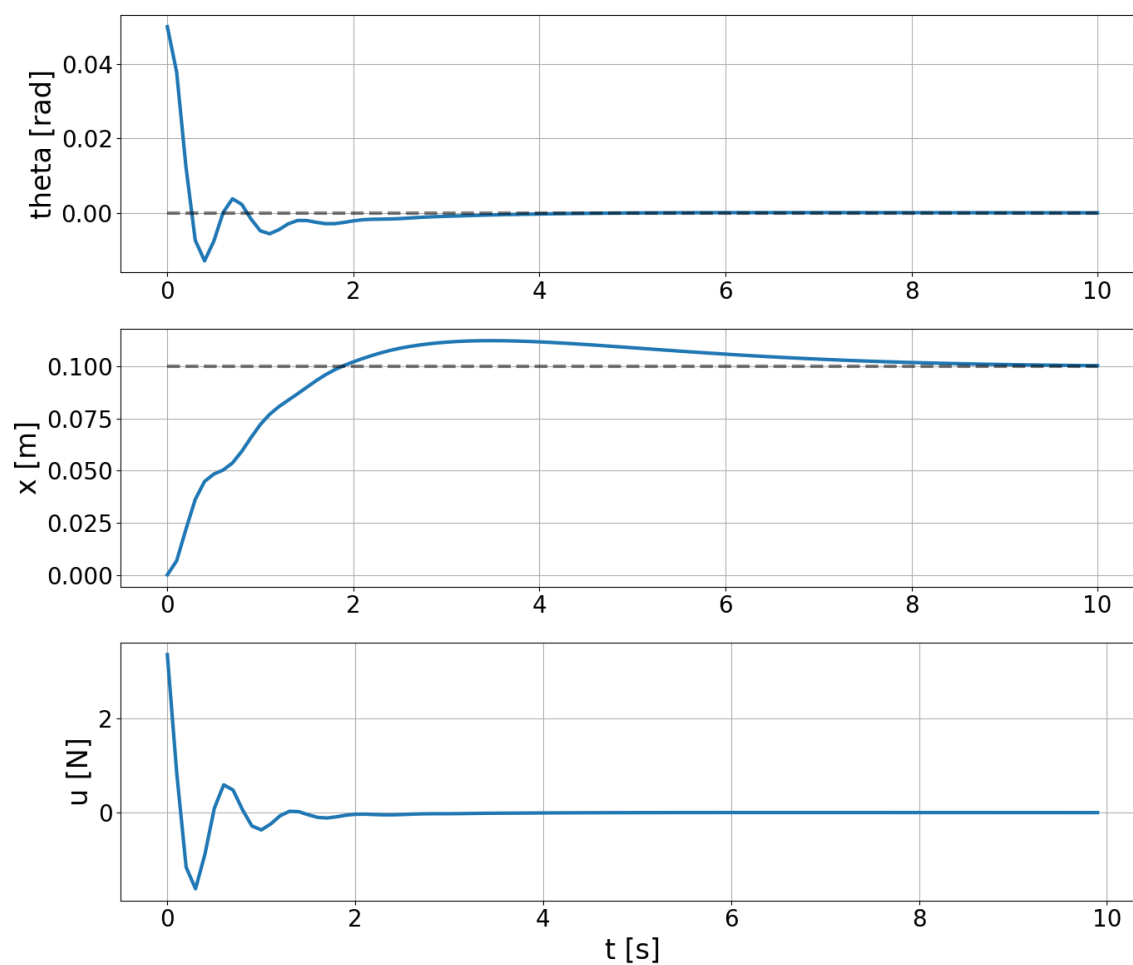
Aby wyeliminować subiektywność strojenia ręcznego, zastosowano algorytm Ewolucji Różnicowej (Differential Evolution), zaimplementowany w module `scipy.optimize`. Zdefiniowano globalną funkcję kosztu J , która gwarantuje „uczciwe” porównanie wszystkich badanych regulatorów:

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

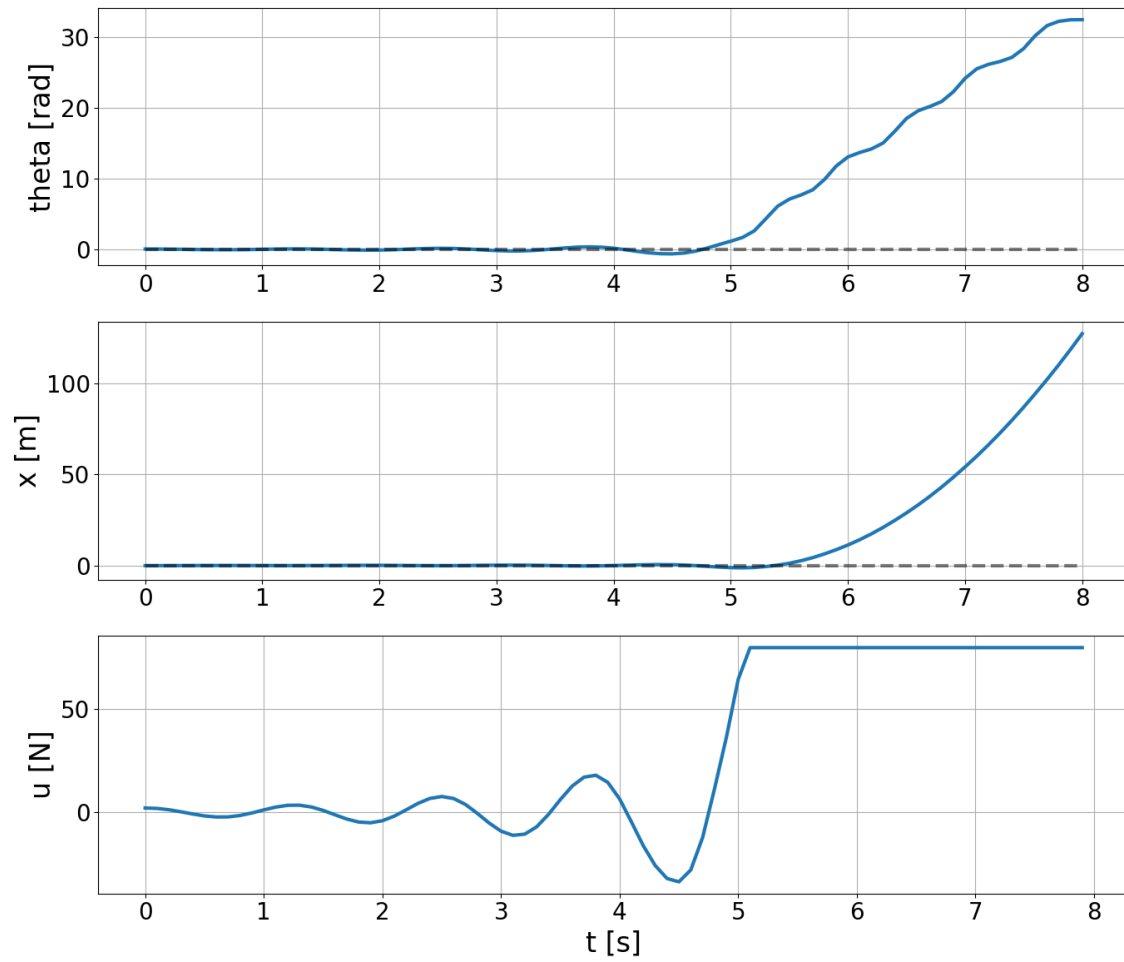
gdzie przyjęto wagi $w_{\theta} = 4.0$ (priorytet stabilizacji), $w_x = 1.0$ (dokładność pozycjonowania) oraz $w_u = 0.01$ (koszt energii). Algorytm operował na populacji 10 osobników przez 20 generacji, co pozwoliło uniknąć minimów lokalnych i znaleźć optymalny zestaw wzmocnień (4.3).

Analiza porównawcza struktur PD i PID W literaturze przedmiotu [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.

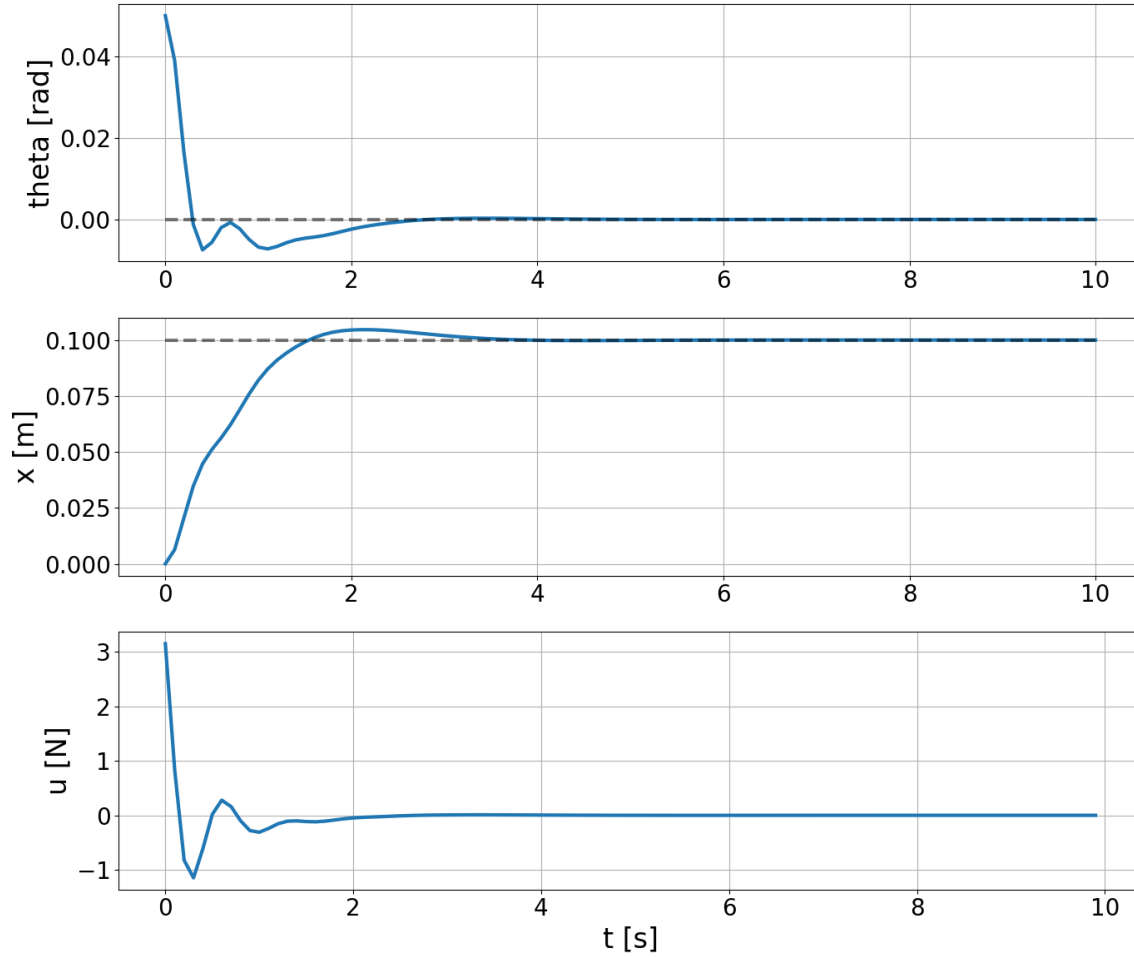
Przeprowadzone badania symulacyjne potwierdziły te wnioski. Próba dodania akcji całkującej ($K_i \neq 0$) skutkowałą zjawiskiem *integral wind-up* – akumulacją błędu w fazie rozruchu, co powodowało przeregulowania wykraczające poza obszar przyciągania stabilnego punktu równowagi. Jak widać na rysunku 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 całkującego (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 [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).

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:

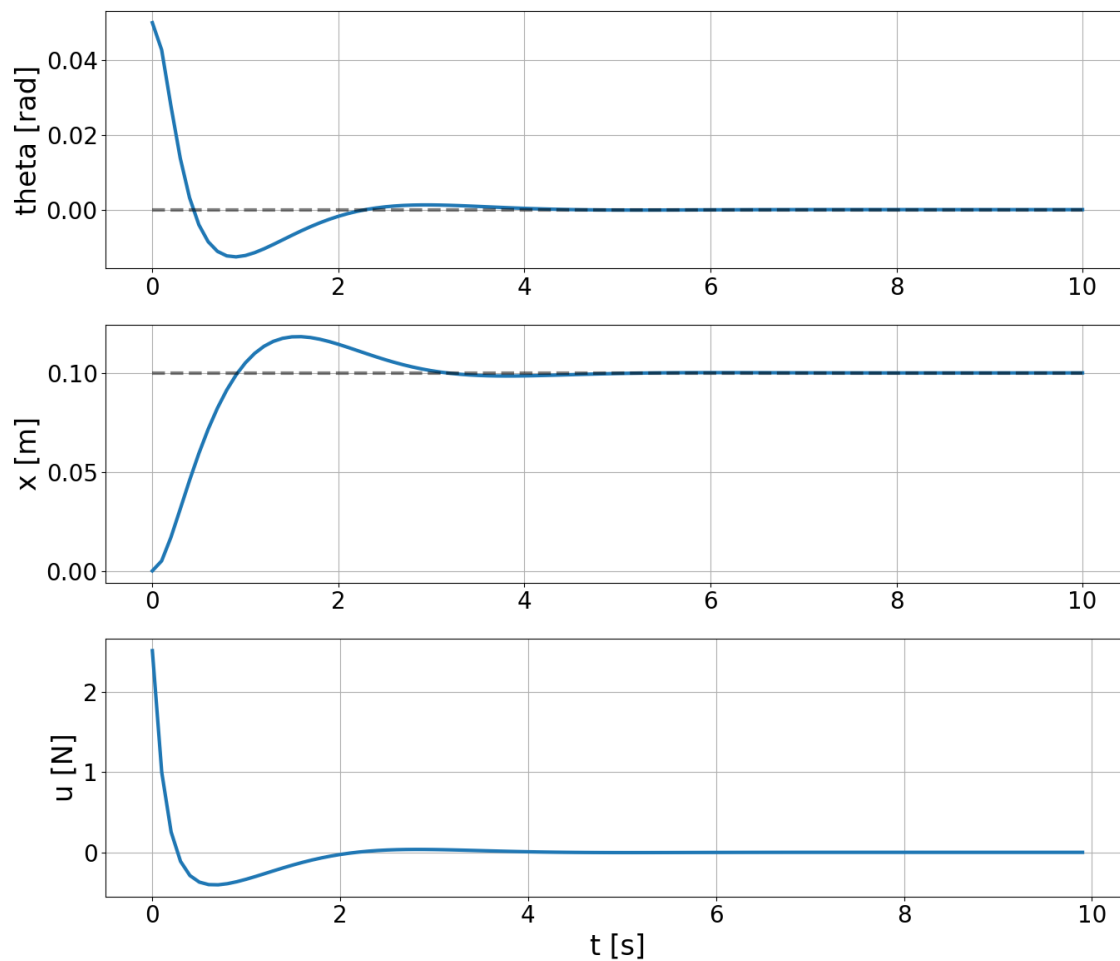
$$Q = \text{diag}([69.44, 76.98, 17.70, 14.17]), \quad R = 8.028. \quad (29)$$

Dodatkowy człon PD na pętli pozycji (zrealizowany analogicznie do wzoru 23) ma na celu poprawę śledzenia skokowych zmian wartości zadanej x_{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

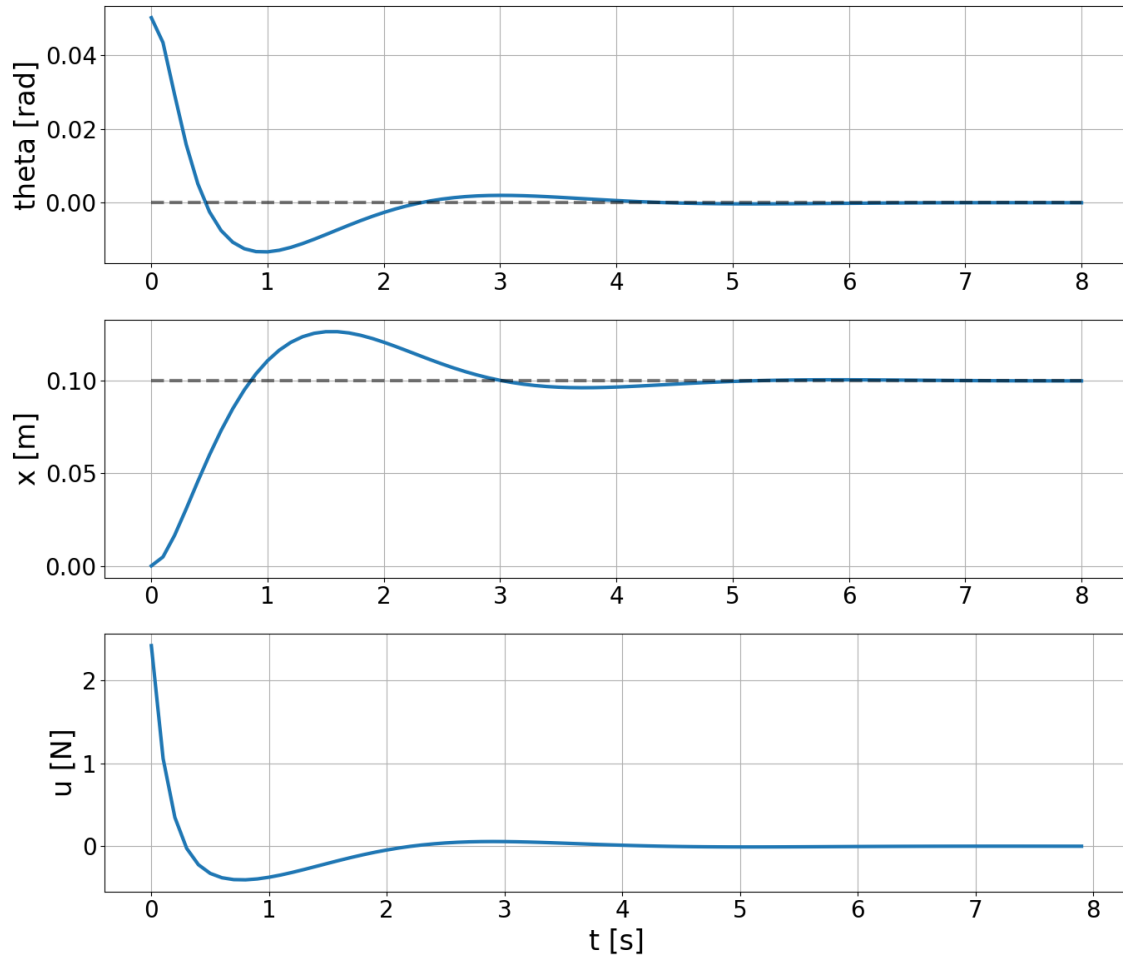
Dobór wag dla regulatora LQR również charakteryzował się ewolucyjnym podejściem do problemu optymalizacji wskaźnika jakości.

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.



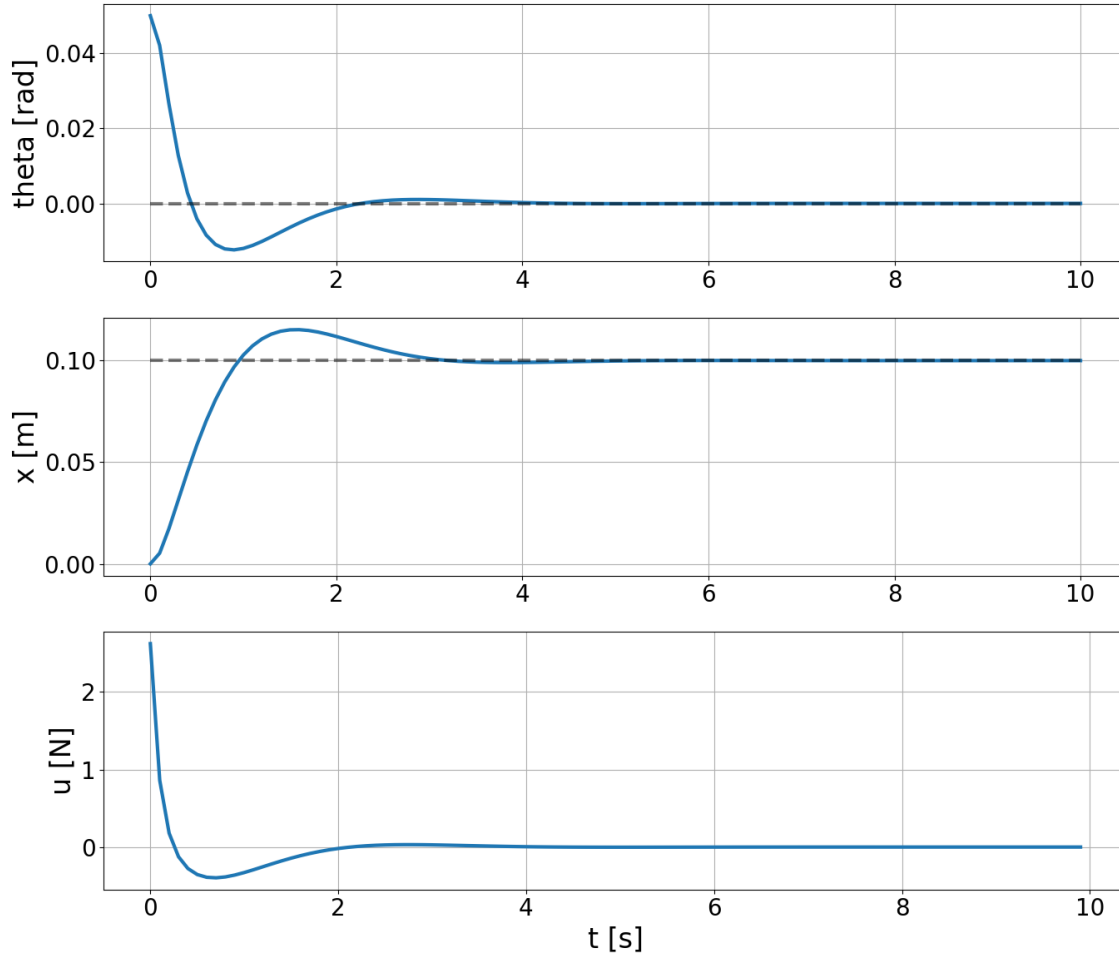
Rysunek 4.4. Próba sterowania LQR z wagami jednostkowymi ($Q = I, R = 1$). Widoczna duża bezwładność układu.

Następnie przeprowadzono strojenie ręczne metodą prób i błędów (zgodnie z regułą Brysona). Ręczne zwiększanie kar za wychylenie kąta (Q_θ) 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 skalar 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 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).

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 (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.

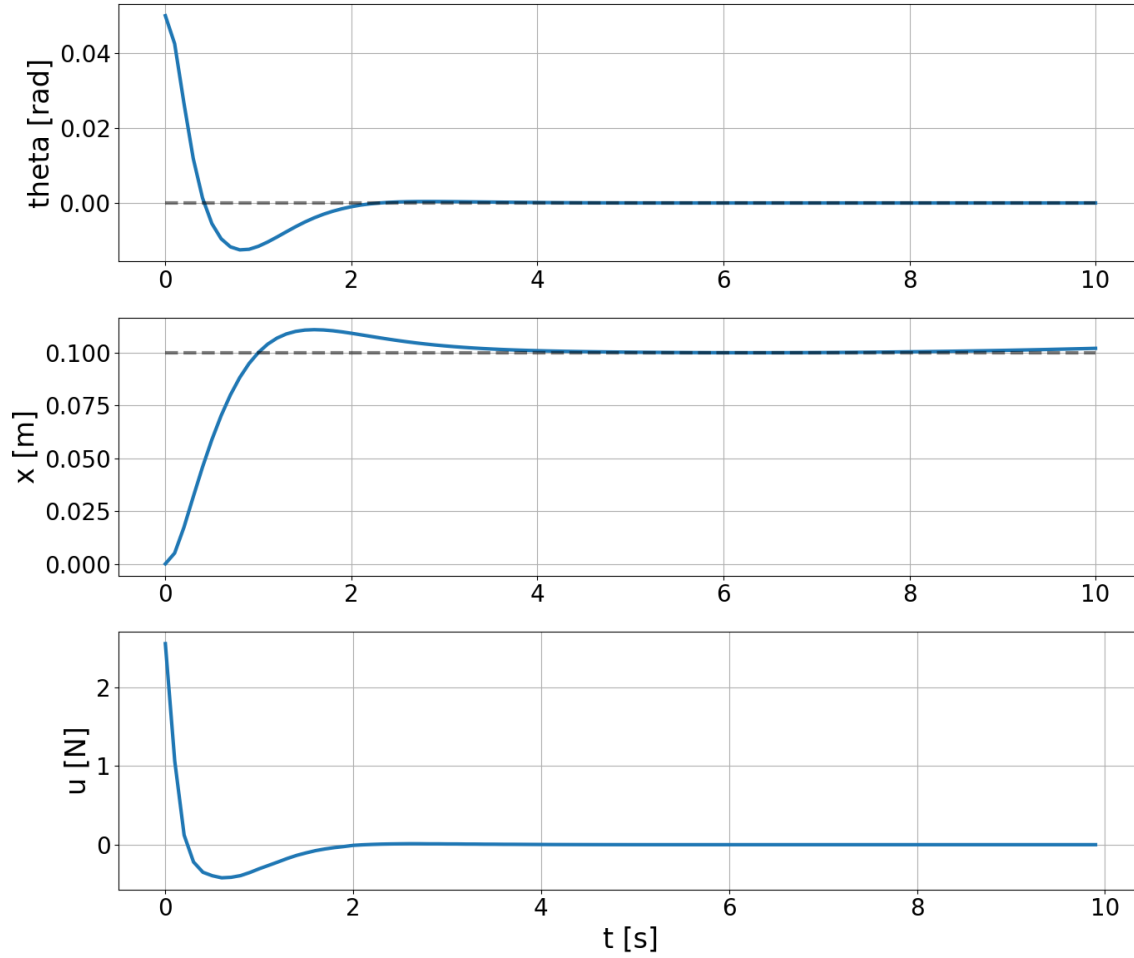
Analiza wykazała, że bezpośrednie przeniesienie macierzy wag Q i R z regulatora LQR do sterownika MPC prowadziło do znaczącego pogorszenia jakości sterowania (wydłużenie czasu regulacji z ok. 3s do ponad 9s). Wynika to z faktu, że model MPC, dzięki jawnemu uwzględnieniu ograniczeń sygnału sterującego,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

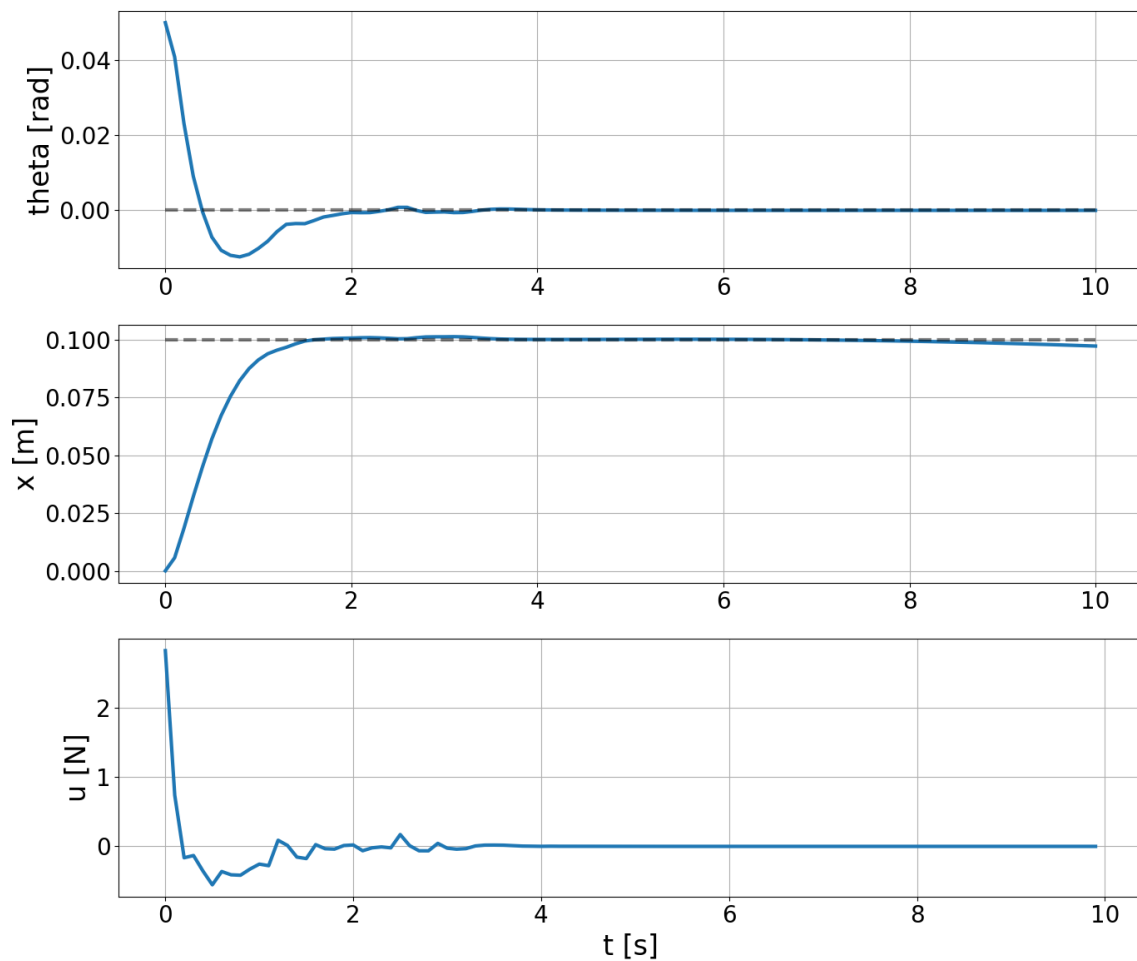
4.4.1. Analiza wpływu kary za energię

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 dostosił 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 [7] oraz [13], 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_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:

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

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”).

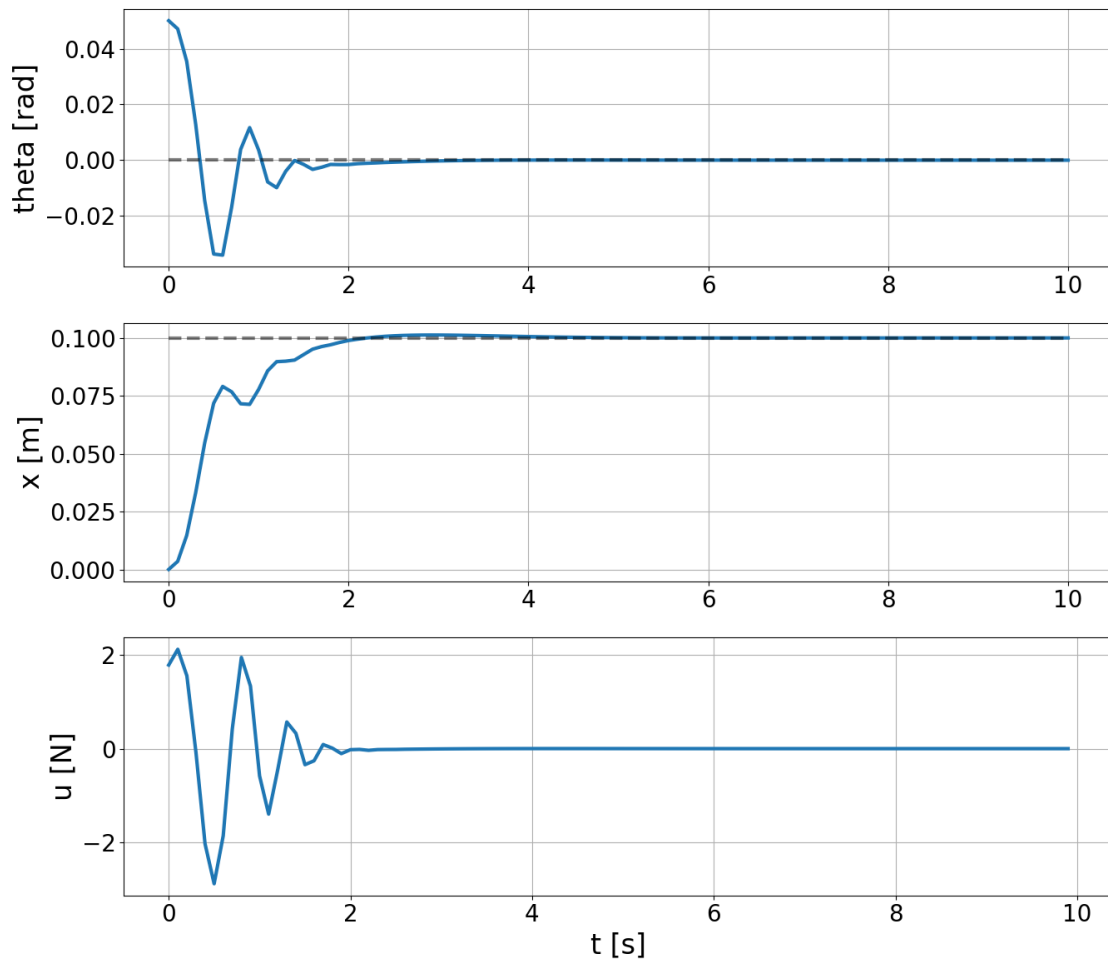
Takie podejście pozwala na rozszerzenie obszaru stabilności regulatora w porównaniu do klasycznego LQR, co potwierdzają wyniki badań w pracy [7].

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.

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 [7], 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 $\Delta t = 0,1 \text{ s}$ 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 silnie duże odchyłki.

$$\text{MSE} = \frac{1}{N} \sum_{k=1}^N (y[k] - y_{\text{ref}}[k])^2 \quad (37)$$

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

$$\text{MAE} = \frac{1}{N} \sum_{k=1}^N |y[k] - y_{\text{ref}}[k]| \quad (38)$$

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

$$\text{ISE} = \int_0^T (y(t) - y_{\text{ref}}(t))^2 dt \quad (39)$$

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

$$\text{IAE} = \int_0^T |y(t) - y_{\text{ref}}(t)| dt \quad (40)$$

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

$$E_{u,L2} = \int_0^T u(t)^2 dt \quad (41)$$

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

$$E_{u,L1} = \int_0^T |u(t)| dt \quad (42)$$

- **Czas regulacji t_s (Settling Time)** – Czas, po którym sygnał wyjściowy trwale wchodzi w kanał tolerancji (przyjęto $\pm\epsilon$ od wartości zadanej) i już go nie opuszcza.
- **Przeregulowanie M_p (Overshoot)** – Maksymalne procentowe odchylenie sygnału od wartości zadanej w odniesieniu do wartości skoku.

$$M_p = \frac{\max(y) - y_{\text{ref}}}{y_{\text{ref}}} \cdot 100\% \quad (43)$$

- **Uchyb ustalony e_{ss} (Steady-state Error)** – Średnia wartość uchybu w końcowej fazie symulacji (ostatnie 10% czasu), określająca dokładność statyczną regulacji.
- **Wskaźnik tłumienia zakłóceń (Robustness/SNR)** – W scenariuszach z wiatrem:

stosunek wartości skutecznej (RMS) błędu regulacji do wartości skutecznej siły zakłócającej.

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 kaskadowy układ PD-PD oraz hybrydowy PD-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 odpowiedź skokowej, pozwalający ocenić szybkość działania (czas regulacji) oraz tłumienie oscylacji przez poszczególne regulatory.

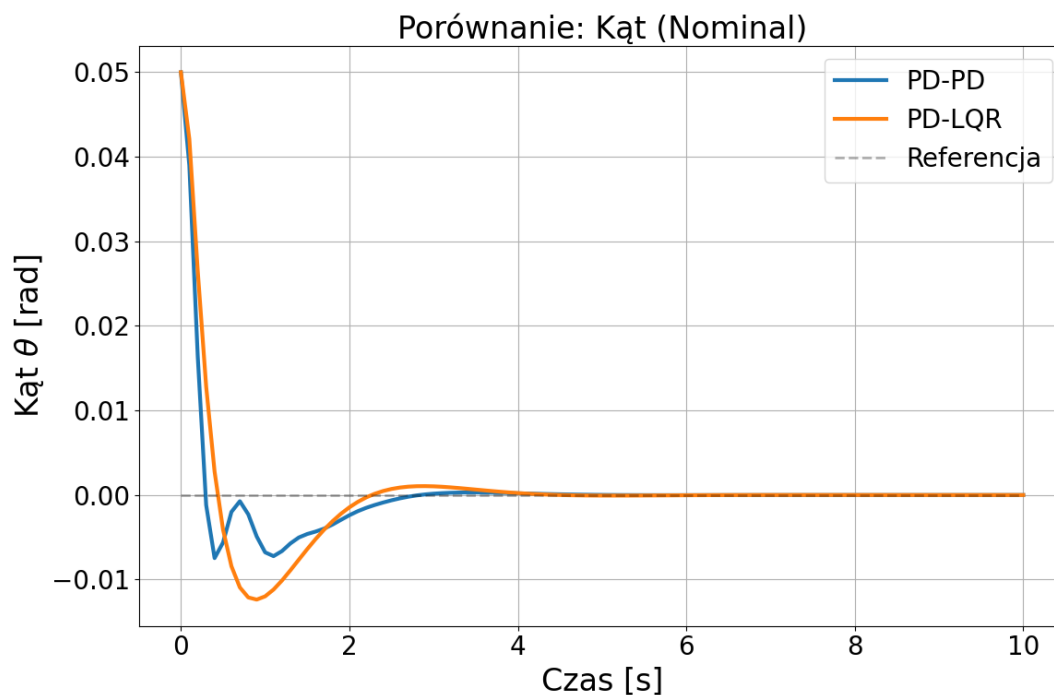
6.1.1. Charakterystyka regulatorów klasycznych

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 wyraźne różnice w filozofii działania obu układów. Regulator PD-LQR, korzystający z pełnego sprzężenia od wektora stanu i macierzy wzmocnień wyznaczonej metodą optymalizacji LQR, charakteryzuje się bardzo pożądanym, aperiodycznym przebiegiem dochodzenia do równowagi. Krzywa ma charakter łagodny, pozbawiony zbędnych oscylacji, co świadczy o wysokim zapasie stabilności i doskonałym tłumieniu.

Zupełnie inaczej zachowuje się regulator kaskadowy PD-PD. Jego odpowiedź, choć stabilna, obarczona jest znacznie dłuższym czasem ustalania się. Widoczne są charakterystyczne oscylacje resztkowe, wynikające z braku bezpośredniego uwzględnienia sprzężeń między dynamiką kątową a liniową w procesie doboru nastaw. Te oscylacje katowe mają

swoje bezpośrednie przełożenie na pozycję wózka, co doskonale widać na Rysunku 6.2. W przypadku regulatora PD-LQR, wózek wykonuje jeden zdecydowany ruch kompensacyjny, po czym płynnie i monotonicznie wraca w okolice punktu zerowego. Cały proces stabilizacji pozycji zamyka się w czasie około 3 sekund. Tymczasem wózek sterowany regulatorem PD-PD "błądzi" wokół zera znacznie dłużej, wykonując szereg korekt przód-tył, będących odpowiedzią na oscylacje wahadła.

Potwierdzeniem wyższej efektywności podejścia LQR jest analiza sygnału sterującego (Rys. 6.3). Regulator hybrydowy generuje sygnał o mniejszej amplitudzie szczytowej i znacznie łagodniejszym przebiegu. Wymaga on zatem mniejszej siły (a co za tym idzie momentu napędowego silnika) do osiągnięcia lepszego efektu końcowego. Jest to dowód na to, że regulator LQR, minimalizując wskaźnik kwadratowy, w sposób naturalny dąży do oszczędzania energii sterowania ($E_u \approx 0.54$), podczas gdy "sztywny" regulator PD-PD zużywa jej niemal dwukrotnie więcej ($E_u \approx 0.85$) na walkę z własnymi oscylacjami.

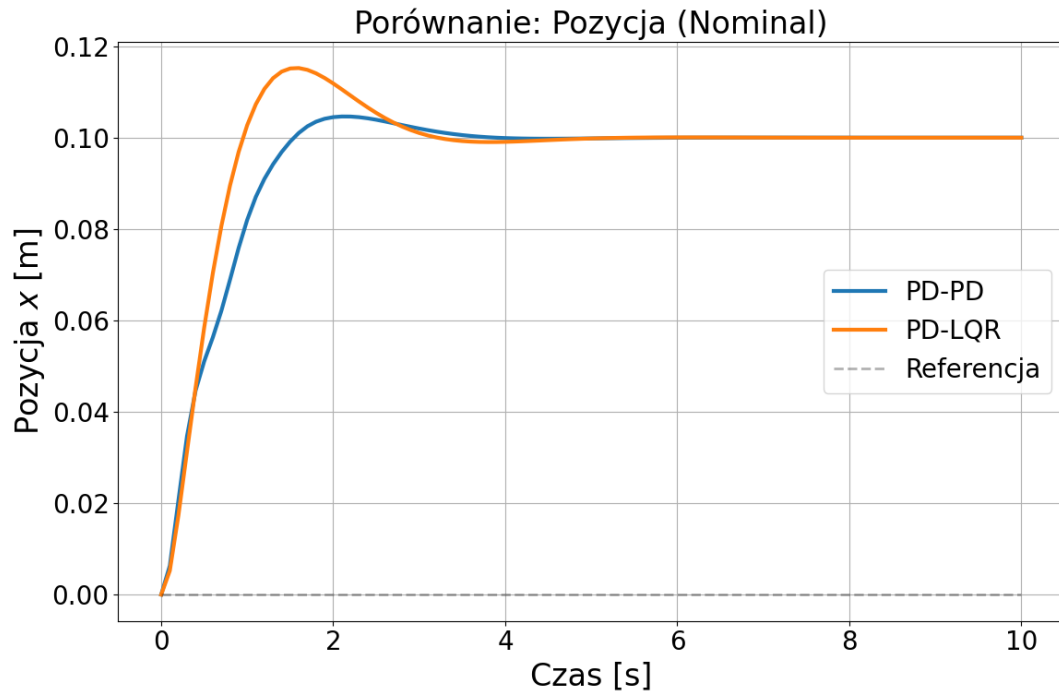


Rysunek 6.1. Przebieg kąta θ dla regulatorów klasycznych (Warunki nominalne).

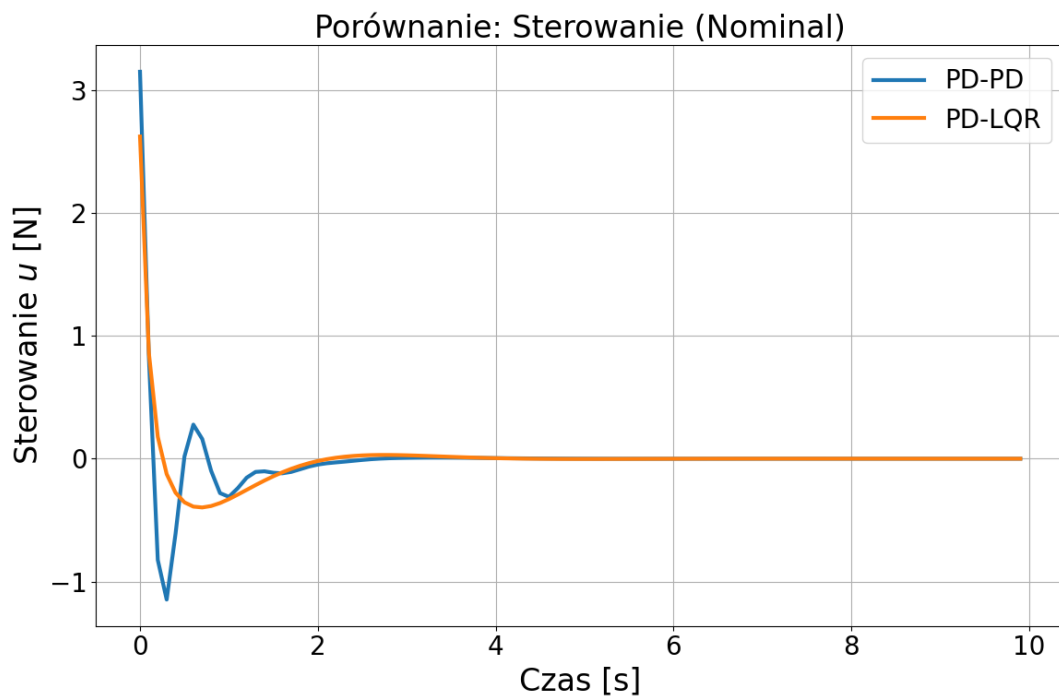
6.1.2. Charakterystyka regulatorów zaawansowanych

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



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



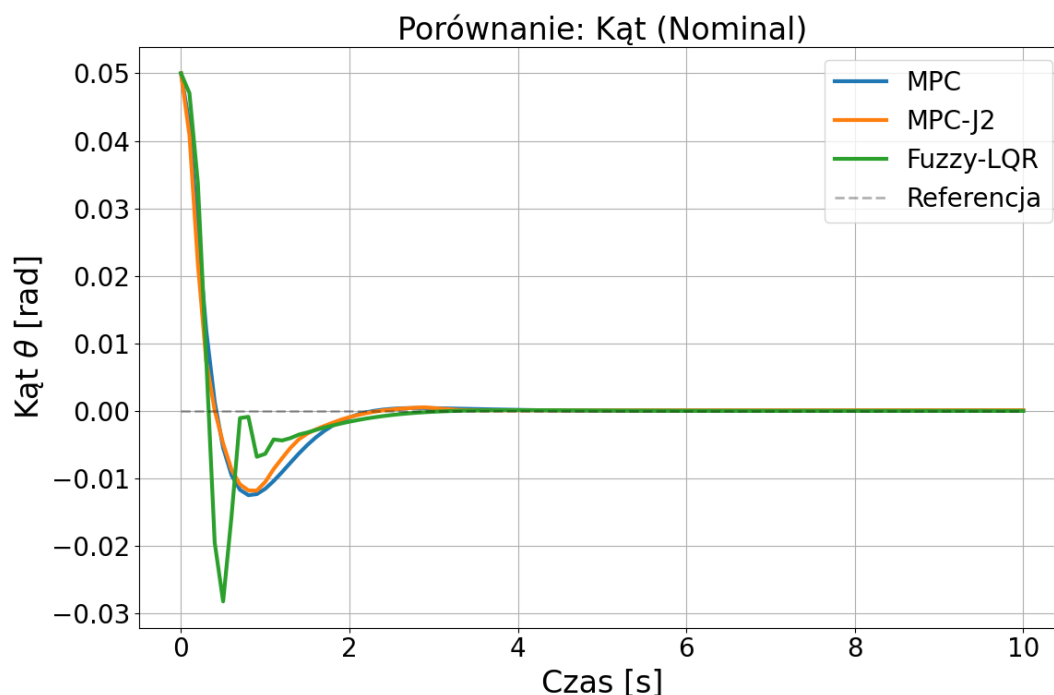
Rysunek 6.3. Sygnał sterujący u dla regulatorów klasycznych (Warunki nominalne).

najszybsze zniwelowanie uchybu kąowego. Dzięki temu podejściu, czas regulacji kąta jest bezkonkurencyjny 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 "zryw", co widać na wykresie pozycji (Rys. 6.5). Choć wózek szybko wraca do zera, początkowe szarpnięcie 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. Działa on z "rozmysłem", planując ruch wózka na kilkanaście kroków do przodu. Jego priorytetem jest znalezienie kompromisu między 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 potrafi 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 zbliżone do optymalnego LQR, co potwierdza wysoką efektywność algorytmów predykcyjnych.

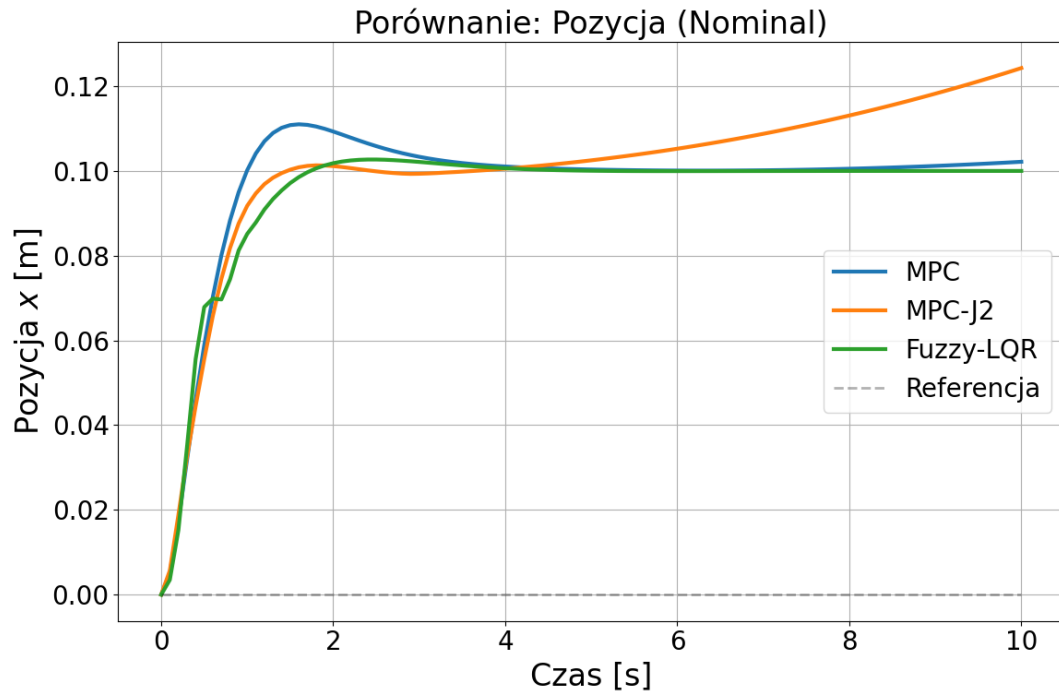
Wariant MPC-J2, mimo zastosowania innej funkcji kosztu, zachowuje się bardzo podobnie do klasycznego MPC w warunkach nominalnych. Różnice między nimi ujawniają się dopiero w teście odpornościowym.



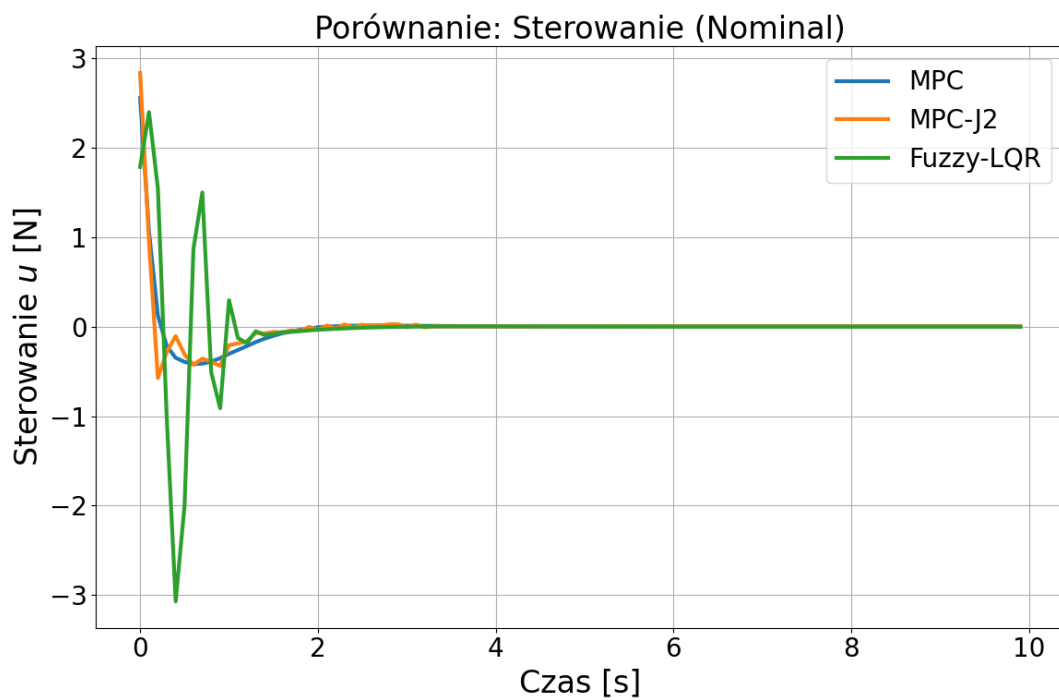
Rysunek 6.4. Przebieg kąta θ dla regulatorów zaawansowanych (Warunki nominalne).

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



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



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

kierunku. Test ten miał na celu sprawdzenie właściwości "robust"(odpornościowych)

regulatorów, czyli ich zdolności do utrzymania stabilności mimo działania nieznanych, zewnętrznych sił.

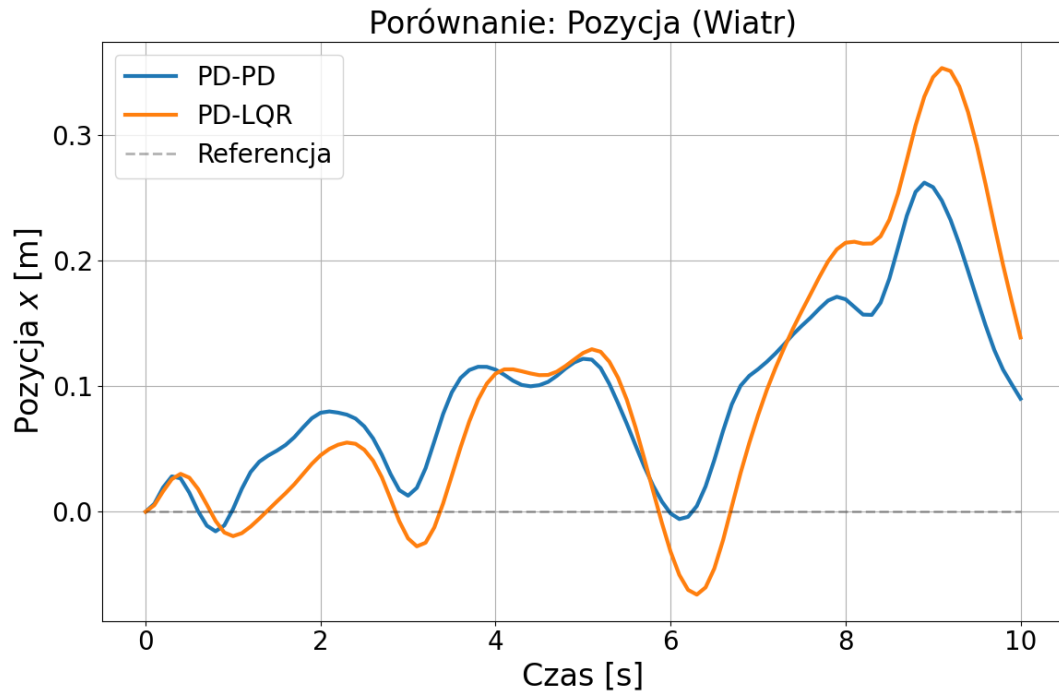
Podstawowym problemem fizycznym w tym scenariuszu jest zjawisko sprzężenia dryfu. Aby skompensować siłę wiatru pchającą wahadło np. w prawo, wózek musi nieustannie przyspieszać w prawo, aby "podjechać" pod środek ciężkości wahadła i wytworzyć moment siły bezwładności przeciwdziałający wiatrowi. Oznacza to, że skuteczna walka 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. ?? i 6.7) oba regulatory radzą sobie poprawnie, choć widoczne są ograniczenia ich struktur. PD-PD dopuszcza do powstawania większych oscylacji kątowych ($MSE_\theta \approx 6.0 \cdot 10^{-4}$), co jest efektem słumiennego reagowania członu różniczkującego na szum. PD-LQR, dzięki lepszemu tłumieniu, utrzymuje wahadło stabilniej, jednak odbywa się to kosztem większego dryfu pozycji ($Max|x| \approx 0.35$ m wobec 0.26 m dla PD-PD).

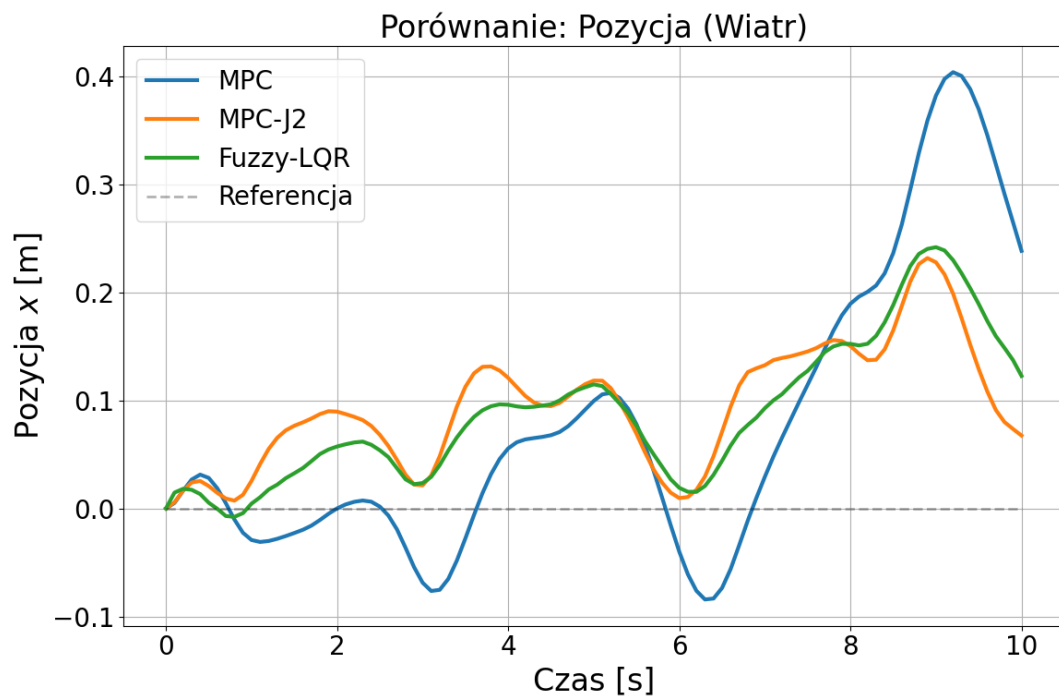
Prawdziwa przepaść jakościowa widoczna jest jednak w grupie zaawansowanej (Rys. ?? i 6.8). Tutaj bezdyskusyjnym zwycięzcą w kategorii precyzji stabilizacji okazał się regulator Fuzzy-LQR. Dzięki nieliniowej strukturze bazy reguł, regulator ten potrafi dynamicznie zmieniać swoje wzmocnienia. Dla małych wychyleń działa łagodnie, ale każda nagła zmiana kąta czy prędkości kątowej wywołana podmuchem wiatru spotyka się z natychmiastową, silną kontrreakcją. Efekt jest spektakularny: wahadło jest trzymane niemal "na sztywno" w pionie, co skutkuje najmniejszym błędem średniokwadratowym ($MSE_\theta \approx 3.0 \cdot 10^{-4}$) oraz, co ciekawe, najmniejszym dryfem pozycji ($Max|x| \approx 0.24$ m). Ponieważ wahadło nie wychyla się mocno, wózek nie musi wykonywać długich rajdów ratunkowych. Ceną za ten wyczyn jest jednak gigantyczne zużycie energii ($E_u \approx 85.9$), wielokrotnie wyższe od konkurencji. Regulator ciągle "szarpie" wózkiem, co w rzeczywistym układzie mogłoby prowadzić do przegrzania napędu.

Zupełnie inną strategię w obliczu wiatru przyjmuje MPC. Ograniczony funkcją kosztu, która karze duże sterowania, MPC zachowuje się bardziej pasywnie. Pozwala on wahadłu na pewne, kontrolowane wychylenia pod wiatr, "płynąć" z zakłóceniem. Skutkuje to większym dryfem wózka ($Max|x| \approx 0.40$ m), który jest po prostu spychany przez wiatr, ale za to zużycie energii jest absolutnie minimalne ($E_u \approx 12.6$). Jest to strategia "ekonomiczna-przetrwać zakłócenie przy minimalnym wysiłku.

Warto odnotować całkowitą porażkę wariantu MPC-J2 w tym teście. Jego funkcja kosztu, karząca bezpośrednio wartość sterowania (u), a nie jej zmianę (Δu), okazała się zbyt restrykcyjna. W obliczu silnego wiatru regulator ten "bał się" użyć większej siły, co doprowadziło do przekroczenia krytycznego kąta wychylenia i przewrócenia się wahadła. Jest to cenna lekcja projektowa, pokazująca jak dobór funkcji celu determinuje odporność układu.



Rysunek 6.7. Dryf pozycji x pod wpływem wiatru – regulatory klasyczne.



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

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).

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

Wskaźnik	PD-PD	PD-LQR	MPC	MPC-J2	Fuzzy-LQR
MSE_{θ}	0.00005	0.00006	0.00006	0.00005	0.00007
IAE_{θ}	0.01684	0.02441	0.02287	0.02010	0.02301
$T_{s,\theta}$ [s]	0.30000	1.30000	1.20000	1.10000	0.70000
MSE_x	0.00040	0.00038	0.00038	0.00046	0.00037
$T_{s,x}$ [s]	1.20000	2.20000	0.90000	1.00000	1.20000
E_u	0.84515	0.54055	0.56337	0.62171	2.84489

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

Wskaźnik	PD-PD	PD-LQR	MPC	MPC-J2	Fuzzy-LQR
MSE_{θ}	0.00060	0.00067	0.00058	0.00067	0.00031
IAE_{θ}	0.18981	0.20099	0.18215	0.20848	0.13488
$Max \theta $ [rad]	0.06821	0.06315	0.06235	0.06577	0.05000
MSE_x	0.00466	0.01147	0.01772	0.00317	0.00440
$Max x $ [m]	0.26214	0.35342	0.40389	0.23177	0.24186
E_u	15.39160	13.49916	12.59235	14.42652	85.88725

6.4. Dyskusja i wnioski końcowe

Przeprowadzone badania symulacyjne, w zestawieniu z literaturą przedmiotu, pozwalają na sformułowanie szeregu istotnych wniosków końcowych. Wykazano, że wybór odpowiedniego algorytmu sterowania dla wahadła odwróconego nie jest kwestią trywialną i zależy ściśle od przyjętych kryteriów projektowych oraz ograniczeń systemowych.

Po pierwsze, wyniki jednoznacznie potwierdzają tezę, że nie istnieje jeden, uniwersalny "super-regulator", który dominowałby we wszystkich aspektach sterowania. Mamy do czynienia z fundamentalnym kompromisem inżynierskim między jakością regulacji a kosztami eksploatacyjnymi.

Precyzja geometryczna a sterowanie rozmyte Jeżeli priorytetem jest bezwzględne utrzymanie punktu pracy, np. w robotyce precyzyjnej lub układach stabilizacji broni, bezkonkurencyjny okazał się system rozmyty (Fuzzy-LQR). Jak zauważają w swojej pracy Roose i in. [13], sterowniki oparte na logice rozmytej wykazują naturalną odporność na nieliniowości i zakłócenia, co znalazło potwierdzenie w eksperymencie z wiatrem (Tabela 6.2). Fuzzy-LQR

potrafił niemal całkowicie zniwelować wpływ losowych podmuchów, utrzymując wahadło w pionie "na sztywno". Jest to zbieżne z obserwacjami Nguyen i Tran [7], którzy wskazują na wyższość hybrydowych układów rozmytych nad klasycznymi metodami liniowymi w trudnych warunkach pracy. Należy jednak pamiętać, że ta precyzja jest okupiona ogromnym wydatkiem energetycznym, co może dyskwalifikować to rozwiązanie w systemach autonomicznych o ograniczonym zasilaniu.

Ekonomia i bezpieczeństwo ruchu (MPC) Z drugiej strony, jeżeli zależy nam na oszczędności energii, płynności ruchu i ochronie mechaniki – co jest kluczowe w pojazdach elektrycznych czy dronach – najlepszym wyborem staje się sterowanie predykcyjne (MPC). Zgodnie z teorią przedstawioną przez Camacho i Bordonsa [9], główną przewagą MPC jest jawne uwzględnianie ograniczeń (w tym przypadku nasycenia sygnału sterującego) w procesie optymalizacji. W naszych badaniach MPC wykazał się "inteligentnym" zarządzaniem zasobami – w obliczu wiatru pozwolił na niewielkie, tymczasowe odchyłki pozycji (dryf), aby uniknąć gwałtownych szarpnięć silnikiem. Potwierdza to wyniki uzyskane przez Jezierskiego i in. [8], którzy również wskazali na wyższą efektywność energetyczną MPC w porównaniu do klasycznego LQR.

Rola rozwiązań klasycznych Nie można również pominąć prostego regulatora PD-LQR. Mimo że ustępuje on rozwiązaniom zaawansowanym w skrajnych sytuacjach, oferuje on bardzo korzystny stosunek jakości do złożoności obliczeniowej. Jak wykazali Varghese i in. [6], LQR pozostaje solidnym standardem przemysłowym, zapewniającym wystarczającą stabilność dla szerokiej klasy obiektów przy minimalnym narzucie implementacyjnym. W naszych testach PD-LQR stanowił doskonały "złoty środek", będąc znacznie lepszym od prostego PD-PD (zgodnie z wynikami Prasad et al. [5]), a jednocześnie tańszym obliczeniowo od MPC.

Podsumowując, projektant systemu sterowania musi świadomie balansować między sztywnością regulacji (Fuzzy) a jej kosztem (MPC). Niniejsza praca dowodzi, że w zależności od definicji funkcji celu – czy jest nią minimalizacja błędu, czy minimalizacja energii – optymalny wybór algorytmu ulega diametralnej zmianie.

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]
φ	– Kąt odchylenia wahadła od pionu [rad]
$\dot{\varphi}$	– 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	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 ($\theta = \pi$).	17
3.1	Przykładowa realizacja stochastycznego procesu zakłócenia (wiatru) działającego na wahadło w czasie symulacji.	20
3.2	Zrzut ekranu z animacji realizowanej w środowisku Python (biblioteka Matplotlib). Widoczny wózek, wahadło oraz zakres ruchu.	21
4.1	Destabilizujący wpływ członu całkującego (PID) - widoczne narastające oscylacje i utrata stabilności.	24
4.2	Stabilna, lecz oscylacyjna praca regulatora PD-PD przy strojeniu ręcznym. . .	25
4.3	Przebiegi czasowe dla zoptymalizowanych nastaw regulatora PD-PD (algorytm Differential Evolution).	26
4.4	Próba sterowania LQR z wagami jednostkowymi ($Q = I, R = 1$). Widoczna duża bezwładność układu.	28
4.5	Wyniki strojenia ręcznego LQR metodą Brysona.	29
4.6	Optymalne przebiegi regulatora PD-LQR po zastosowaniu algorytmu genetycznego.	30
4.7	Zoptymalizowany regulator MPC ($N_p = 12$) - szybka i gładka stabilizacja. . . .	32
4.8	Optymalny kompromis między jakością regulacji a energią w MPC-J2.	33
4.9	Efektywne sterowanie Fuzzy-LQR po optymalizacji bazy reguł.	35
6.1	Przebieg kąta θ dla regulatorów klasycznych (Warunki nominalne).	40
6.2	Przebieg pozycji x dla regulatorów klasycznych (Warunki nominalne).	41
6.3	Sygnał sterujący u dla regulatorów klasycznych (Warunki nominalne).	41

6.4	Przebieg kąta θ dla regulatorów zaawansowanych (Warunki nominalne). . . .	42
6.5	Przebieg pozycji x dla regulatorów zaawansowanych (Warunki nominalne). .	43
6.6	Sygnał sterujący u dla regulatorów zaawansowanych (Warunki nominalne). .	43
6.7	Dryf pozycji x pod wpływem wiatru – regulatory klasyczne.	45
6.8	Dryf pozycji x pod wpływem wiatru – regulatory zaawansowane.	45

Spis tabel

3.1	Parametry fizyczne modelu przyjęte w symulacji	19
6.1	Wskaźniki jakości (Kąt i Pozycja) - warunki nominalne	46
6.2	Wskaźniki jakości (Kąt i Pozycja) - zakłócenia wiatrem	46