

Politechnika Warszawska

WYDZIAŁ ELEKTRONIKI
I TECHNIK INFORMACYJNYCH



Instytut 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
Robert Nebeluk

WARSZAWA 2025

Efektywny układ stabilizacji odwróconego wahadła na wózku

Streszczenie.

Effective stabilisation system of the inverted pendulum on the cart

Abstract.

Spis treści

1. Wstęp	7
1.1. Cel i zakres pracy	8
1.2. Przegląd literatury	9
2. Model matematyczny układu	10
2.1. Opis fizyczny i założenia upraszczające	10
2.2. Analiza kinematyczna	10
2.3. Równania dynamiki układu	11
2.3.1. Bilans sił w kierunku poziomym	11
2.3.2. Bilans momentów sił	12
2.4. Nieliniowy model w przestrzeni stanów	12
2.5. Liniaryzacja modelu w punkcie pracy	13
3. Środowisko symulacyjne i implementacja	15
3.1. Narzędzia programistyczne	15
3.2. Konfiguracja symulacji	15
3.3. Modelowanie zakłóceń	16
3.4. Wizualizacja i animacja	17
Bibliografia	19
Wykaz symboli i skrótów	20
Spis rysunków	20
Spis tabel	20

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

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 i LQR, 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 realizacja środowiska symulacyjnego oraz przeprowadzenie wielokryterialnej analizy porównawczej wybranych algorytmów sterowania dla nieliniowego układu odwróconego wahadła na wózku. Praca koncentruje się na zestawieniu klasycznych metod regulacji (PID, LQR) z zaawansowanymi technikami sterowania predykcyjnego (MPC) oraz rozmytym.

Główny nacisk położono na praktyczną ocenę różnic w działaniu poszczególnych regulatorów, ze szczególnym uwzględnieniem kompromisu między jakością stabilizacji a wydatkiem energetycznym i złożonością obliczeniową.

W ramach projektu zaimplementowano i poddano badaniom następujące struktury:

1. **Regulator PID** — reprezentujący klasyczne podejście inżynierskie. Układ składa się z dwóch pętli sprzężenia zwrotnego (stabilizacja kąta i pozycji), wyposażonych w mechanizmy zapobiegające nasyceniu członów całkujących (ang. *anti-windup*) oraz ograniczenia sygnału sterującego.
2. **Hybrydowy układ PID-LQR** — struktura łącząca prostotę PID (w pętli regulacji pozycji) z optymalnym regulatorem stanu LQR (w pętli stabilizacji wahadła). Metoda ta stanowi punkt odniesienia dla oceny skuteczności metod bazujących na modelu liniowym.
3. **Regulator MPC** — wariant podstawowy sterowania predykcyjnego z kwadratową funkcją kosztu. Służy on w pracy jako nowoczesny *benchmark*, pozwalający ocenić, jakie korzyści daje uwzględnienie ograniczeń sterowania i dynamiki układu bezpośrednio w procesie optymalizacji.
4. **Regulator MPC z rozszerzonym wskaźnikiem jakości** — wariant badawczy metody predykcyjnej, w którym przeanalizowano wpływ dodatkowych kar w funkcji celu (za gwałtowne zmiany sterowania oraz jego amplitudę) na płynność regulacji i zużycie energii.
5. **Regulator rozmyty (Fuzzy Logic)** — metoda sterowania inteligentnego, wykorzystująca zbiór reguł wnioskowania, stanowiąca alternatywę dla metod analitycznych.

Kluczowym elementem pracy jest weryfikacja działania regulatorów w zadaniu stabilizacji wahadła w pozycji pionowej (punkt pracy) w obecności zakłóceń zewnętrznych. W celu obiektywnego porównania metod przyjęto zestaw wskaźników ilościowych. Jakość stabilizacji weryfikowana jest w oparciu o metryki całkowite błędu (ISE, IAE) dla kąta wychylenia oraz pozycji wózka. Równolegle ocenie poddano ekonomię sterowania, wyznaczając koszt energetyczny poprzez normy L_1 i L_2 sygnału sterującego. Całość uzupełnia analiza odporności układu na zakłócenia (szumy, siły zewnętrzne), 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 [1] 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 [2]. 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 [3] 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. [4] 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 [5] oraz Tatjewskiego [6]. 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 [7], [8]. Natomiast w obszarze sterowania rozmytego cennym źródłem wiedzy metodycznej są opracowania dotyczące modeli Takagi–Sugeno i ich porównań z podejściami klasycznymi [9].

Podsumowując, analiza literatury wskazuje na ewolucję podejść sterowania: od klasycznych paradygmatów PID i LQR [1], przez ujęcia predykcyjne [2], [6], aż po zaawansowane metody hybrydowe i inteligentne [3]. Niniejsza praca wpisuje się w ten nurt badawczy, zestawiając te różnorodne metody w ramach jednego projektu symulacyjnego.

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 [1].

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 [1]:

- 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],
- $u(t)$ — 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 [1]:

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 [1]:

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

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

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:

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

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

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):

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

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

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 [1].

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

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

2. Model matematyczny układu

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

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

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

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

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 [1], równanie momentów przyjmuje postać:

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

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 działa poziomo, stąd jej ramię siły względem punktu obrotu wynosi $l\cos\theta$.

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

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

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 [1]:

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

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):

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

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 [1]:

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

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

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. Liniaryzacja modelu w punkcie pracy

W celu zastosowania algorytmów sterowania liniowego (PID, LQR), przeprowadzono liniaryzację 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 $\dot{\mathbf{x}} = \mathbf{A}\mathbf{x} + \mathbf{B}u + \mathbf{B}_{dist}F_w$ opisują macierze wyznaczone zgodnie z literaturą [1]:

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

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

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

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

2. Model matematyczny układu

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 [1].

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. Kluczowy fragment implementacji algorytmu przedstawiono na Listingu 1.

Listing 1. Implementacja metody Rungego-Kutty 4. rzędu

```
def rk4_step(f, x, u, pars, dt):  
    k1 = f(x, u, pars)  
    k2 = f(x + 0.5 * dt * k1, u, pars)  
    k3 = f(x + 0.5 * dt * k2, u, pars)  
    k4 = f(x + dt * k3, u, pars)  
    return x + (dt / 6.0) * (k1 + 2*k2 + 2*k3 + k4)
```

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 jednocze-

śnie 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	2,4	kg
Masa wahadła	m	0,23	kg
Długość wahadła	l	0,36	m
Przyspieszenie ziemskie	g	9,81	m/s ²
Ograniczenie sterowania	u_{max}	100,0	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):

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

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

Wartości zadane: Celem układu jest osiągnięcie stanu $\mathbf{x}_{ref} = [0, 0, x_{ref}, 0]^T$, gdzie x_{ref} (np. 0,1 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. Model zakłócenia $F_w(t)$ oparty jest na procesie stochastycznym:

1. Generowany jest biały szum gaussowski o zadanej mocy.
2. Sygnał jest wygładzany filtrem uśredniającym (splot z oknem prostokątnym), co pozwala uzyskać bardziej realistyczne, ciągłe w czasie przebiegi siły wiatru, zamiast nieskorelowanego szumu.

W eksperymentach z zakłóceniami, siła F_w jest dodawana bezpośrednio do równań dynamiki w każdym kroku całkowania. Przykładowy przebieg wygenerowanego sygnału zakłócającego przedstawiono na Rys. 3.1, a sposób jego generacji w kodzie źródłowym na Listingu 2.

Listing 2. Klasa generatora zakłóceń wiatru

```
class Wind:
```

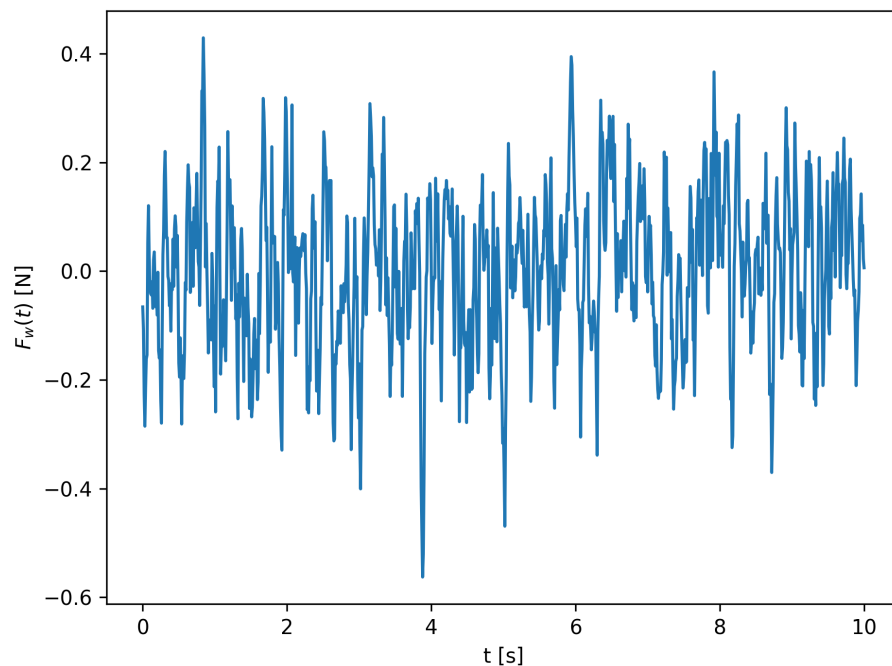


```

def __init__(self, t_end: float, seed=23341, Ts=0.1, power=1e-3, smooth=5):
    rng = np.random.default_rng(seed)
    self.tgrid = np.arange(0.0, t_end + Ts, Ts)
    sigma = np.sqrt(power / Ts)
    w = rng.normal(0.0, sigma, size=self.tgrid.shape)
    if smooth and smooth > 1:
        kernel = np.ones(smooth) / smooth
        self.Fw = np.convolve(w, kernel, mode='same')
    else:
        self.Fw = w

def __call__(self, t: float) -> float:
    return float(np.interp(t, self.tgrid, self.Fw))

```



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 $\theta(t)$.

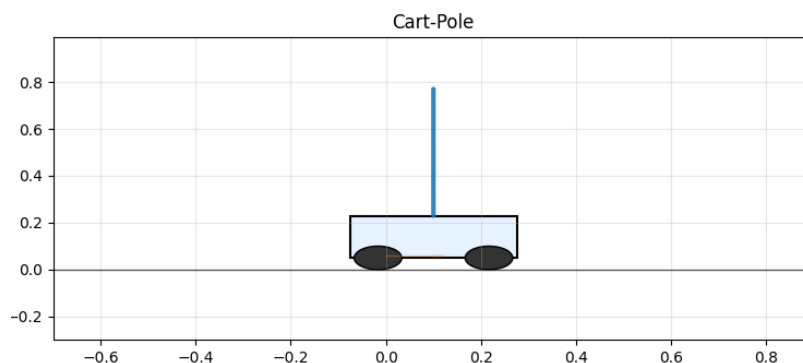
Kluczowym elementem implementacji jest funkcja aktualizująca `update`, wywoływana dla każdego kroku czasowego. Odpowiada ona za przeliczenie współrzędnych kinematycznych (Listing 3) oraz przesunięcie okna widoku kamery tak, aby wózek znajdował się zawsze w centrum, co pozwala na obserwację ruchu na długim dystansie. Dodatkowo rysowany jest „ślad” (ang. *trail*) przebytej drogi przez oś wózka, co ułatwia wizualną ocenę stabilności pozycji.

Listing 3. Logika aktualizacji klatki animacji

```
def pole_end(i):
    cx = x[i]; cy = wheel_r + cart_h
    # Kinematyka prosta wahadla
    px = cx + pole_len * np.sin(th[i])
    py = cy + pole_len * np.cos(th[i])
    return cx, cy, px, py

def update(i):
    cx, cy, px, py = pole_end(i)
    # Aktualizacja pozycji obiektów graficznych
    cart.set_x(cx - cart_w/2)
    wheel1.center = (cx - cart_w/3, wheel_r)
    pole_line.set_data([cx, px], [cy, py])
    # Centrowanie kamery na wozku
    ax.set_xlim(cx - pad, cx + pad)
    return cart, wheel1, pole_line
```

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



Rysunek 3.2. Zrzut ekranu z animacji realizowanej w środowisku Python (biblioteka Matplotlib). Widoczny wózek, wahadło oraz zakres ruchu.

Bibliografia

- [1] 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
- [2] 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
- [3] 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
- [4] 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
- [5] E. F. Camacho i C. Bordons, *Model Predictive Control*, 2 wyd. London: Springer, 2007. DOI: 10.1007/978-1-84628-615-5
- [6] P. Tatjewski, *Sterowanie zaawansowane obiektów przemysłowych. Struktury i algorytmy. Wydanie drugie zmienione*. Akademicka Oficyna Wydawnicza EXIT Andrzej Lang, 2016.
- [7] 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
- [8] 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
- [9] 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

Wykaz symboli i skrótów

x – Położenie wózka [m]

\dot{x} – Prędkość wózka [m/s]

φ – Kąt odchylenia wahadła od pionu [rad]

$\dot{\varphi}$ – Prędkość kątowna 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

- 3.1 Przykładowa realizacja stochastycznego procesu zakłócenia (wiatru)
działającego na wahadło w czasie symulacji. 17
- 3.2 Zrzut ekranu z animacji realizowanej w środowisku Python (biblioteka
Matplotlib). Widoczny wózek, wahadło oraz zakres ruchu. 18

Spis tabel

- 3.1 Parametry fizyczne modelu przyjęte w symulacji 16