

POLITECHNIKA WROCŁAWSKA
WYDZIAŁ ELEKTRONIKI

KIERUNEK: Automatyka i Robotyka (AIR)
SPECJALNOŚĆ: Robotyka (ARR)

**PRACA DYPLOMOWA
INŻYNIERSKA**

Algorytmy sterowania odwróconym
podwójnym wahadłem

Control algorithms of double inverted
pendulum

AUTOR:
Piotr Bogdoł

PROWADZĄCY PRACĘ:
Dr inż. Robert Muszyński,
Katedra Cybernetyki i Robotyki

OCENA PRACY:

Spis treści

1	Wstęp	3
2	Preliminaria matematyczne	5
2.1	Formalizm Lagrange'a	5
2.2	System nieliniowy wejściowo afiniczny	6
2.3	Regulator LQR	6
2.3.1	Równanie stanu i przybliżenie liniowe układu	6
2.3.2	Regulator LQR	7
2.3.3	Dobór wartości parametrów macierzy Q i R	8
2.4	Model chmury	8
2.4.1	Rodzaje chmur	9
2.4.2	Generowanie chmury	9
2.4.3	Tworzenie reguł opartych o model chmury	12
2.5	Generowanie trajektorii ruchu	14
3	Podwójne odwrócone wahadło	17
3.1	Wyprowadzenie równań ruchu	17
3.2	Przekształcenie do postaci afinicznej	19
3.3	Uwzględnienie tarcia	19
4	Sterowanie do punktu	21
4.1	Regulator LQR	21
4.1.1	Przykład zastosowania regulatora LQR	22
4.1.2	Badanie właściwości regulatora LQR	23
4.1.3	Podsumowanie	27
4.2	Regulator LQR z regulatorem PID	27
4.2.1	Porównanie własności regulatora LQR oraz rozszerzonego regulatora LQR	28
4.2.2	Badanie właściwości rozszerzonego regulatora LQR	28
4.2.3	Podsumowanie	30
4.3	Zastosowanie modelu chmury do zmiany nastaw PID	30
4.3.1	Przykład zastosowania rozszerzonego regulatora LQR wykorzystującego model chmury	32
4.3.2	Badanie własności rozszerzonego regulatora LQR wykorzystującego model chmury	34
4.3.3	Podsumowanie	37
4.4	Podsumowanie	37

5	Generowanie i śledzenie trajektorii	39
5.1	Generowanie trajektorii ruchu	39
5.2	Śledzenie trajektorii z wykorzystaniem rozszerzonego regulatora feedforward	40
5.2.1	Algorytm śledzenia trajektorii	40
5.2.2	Zastosowanie rozszerzonego regulatora feedforward do śledzenia trajektorii	41
5.2.3	Badanie własności rozszerzonego regulatora feedforward	43
5.3	Podsumowanie	45
6	Podsumowanie i wnioski	47
	Literatura	49
	Spis tabel	51
	Spis rysunków	53

Do składu pracy wykorzystano system przygotowania dokumentów \LaTeX , opracowany przez L. Lamport [Lam94], będący nakładką systemu \TeX . [Knu86a,Knu86b]. Matematyczne czcionki o nazwie AMS Euler, których używamy w tej pracy, zostały przygotowane przez H. Zapfa [KZ86], przy współpracy z D. Knuthem i jego studentami, na zlecenie Amerykańskiego Towarzystwa Matematycznego. Wybrane czcionki składu tekstu, Antykwa Toruńska [Now97] – jeden z nielicznych krojów pisma zaprojektowany specjalnie dla języka polskiego w sposób uwzględniający jego rytm – w odczuciu autora doskonale współgrają z kształtem czcionki AMS Euler, pozwalając na uzyskanie harmonijnej całości. Składu bezszeryfowego tekstu maszynowego dokonano z użyciem opracowanej przez R. Leviena czcionki o nazwie Inconsolata [Lev15]^a.

[Knu86a] D. E. Knuth, The \TeX book, volume A of Computers and Typesetting. Addison-Wesley, Reading, 1986.

[Knu86b] D. E. Knuth, \TeX : The Program, volume B of Computers and Typesetting. Addison-Wesley, Reading, 1986.

[KZ86] D. E. Knuth i H. Zapf, AMS Euler — A new typeface for mathematics. Scholarly Publishing, 20:131–157, 1986.

[Lam94] L. Lamport, \LaTeX : A Document Preparation System. Addison–Wesley, Reading, 1994.

[Lev15] R. Levien, Inconsolata. <https://levien.com/type/myfonts/inconsolata.html>, 2015.

[Now97] J. Nowacki, Antykwa Toruńska — od początku do końca polska czcionka. *Biuletyn Polskiej Grupy Użytkowników Systemu \TeX* , 9:26–27, 1997.

^a Chyba warto takie informacje szerzyć

Rozdział 1

Wstęp

Czyż nie każdy z nas choć raz w życiu próbował balansować różnego rodzaju podłużnymi przedmiotami postawionymi na dłoni czy palcu tak, aby utrzymać je w pionie? Dla pojedynczego przedmiotu nie jest to wcale łatwe, a co dopiero, jeżeli połączylibyśmy dwa albo nawet trzy takie przedmioty przy użyciu przegubów. Szybko okazałoby się, że jest to, dla człowieka w zasadzie zadanie niewykonalne. Mimo to istnieją jednak różnego rodzaju algorytmy pozwalające na sterowanie takimi obiektami w efektywny sposób i tym właśnie się w tej pracy zajmujemy.

Odwrócone wahadło jest niestabilnym, wielowymiarowym systemem nieliniowym, które w procesie rozwoju teorii sterowania oraz metod sterowania stało się idealnym obiektem służącym do testowania nowych teorii i sposobów sterowania [QDFY11]. Odwrócone wahadło składa się z pręta połączonego swobodnie przegubem obrotowym z poruszającym się poziomo, napędzanym wózkiem. Dzięki stosunkowo prostemu modelowi matematycznemu można w łatwy sposób symulować zachowanie takiego odwróconego wahadła. Pozwala to na implementację różnego rodzaju algorytmów w środowiskach symulacyjnych, weryfikacji ich przydatności do sterowania obiektem nieliniowego, badanie ich własności, porównywanie oraz ulepszanie. Odwrócone wahadło może być też dobrym i dość widowiskowym przykładem, pokazującym jak trudne jest sterowanie nieliniowymi i niedosterowanymi obiektami. Zainteresowaniem badaczy cieszy się również odwrócone, podwójne czy też nawet potrójne wahadło. Istnieje wiele prac opisujących różnego rodzaju algorytmy stosowane do sterowania takimi wahadłami. Dla przykładu możemy przytoczyć [Raj16], w której wykorzystano jedną z metod maszynowego uczenia, uczenie przez wzmacnianie, do stabilizowania podwójnego, odwróconego wahadła. Kolejnym przykładem może być [TNY03], w której użyto sztucznej sieci neuronowej do sterowania podwójnym, odwróconym wahadłem. Innym podejściem do problemu, opisanym w [KCHI17], jest zastosowanie regulatora LQR do stabilizacji podwójnego, odwróconego wahadła. W [QDFY11] natomiast wykorzystano sterownik predykcyjny z dyskretnym czasem sterowania. Oczywiście przytoczone tu prace są tylko niewielką częścią różnego rodzaju rozwiązań stosowanych do sterowania odwróconym wahadłem.

Celem pracy jest przegląd nowoczesnych algorytmów sterowania stosowanych do odwróconego, podwójnego wahadła. W pracy zebrano przykładowe algorytmy służące temu celowi oraz opisano sposób ich implementacji w środowisku symulacyjnym. Następnie wykonano badania symulacyjne własności algorytmów oraz

porównano je. Do badań wybrano algorytmy opisane w [LZG16] oraz [GAK13].

Jako pierwsze w pracy przedstawimy krótkie wprowadzenie dotyczące zagadnień matematycznych oraz z zakresu teorii sterowania, związanych z pojęciami wykorzystanymi w dalszych rozdziałach. W kolejnym rozdziale wyprowadzimy model matematyczny podwójnego, odwróconego wahadła. Wykorzystując formalizm Lagrange'a, wyliczymy równania ruchu dla naszego wahadła i podamy ich postać wektorową oraz afiniczną. W rozdziale zostaną rozpatrzone dwa modele, pierwszy bez uwzględniania tarcia i drugi z jego uwzględnieniem. Rozdział 4 poświęcony jest sterowaniu do punktu podwójnego, odwróconego wahadła. Przedstawimy trzy sposoby sterowania: z wykorzystaniem regulatora LQR, regulatora LQR połączonego z regulatorem PID oraz regulatora LQR z regulatorem PID, którego nastawy są modyfikowane z wykorzystaniem modelu chmury. Wykonie symulacji układów regulacji pozwoli na zbadanie ich własności. W rozdziale 5 przedstawimy sposób śledzenia trajektorii przez podwójne, odwrócone wahadło. Pokażemy, w jaki sposób można generować trajektorie ruchu wahadła oraz wykorzystamy regulator feedforward wraz z regulatorem LQR do jej śledzenia. Ostatni rozdział pracy został przeznaczony krótkiemu podsumowaniu oraz przedstawieniu spostrzeżeń i wniosków, które nasunęły się autorowi podczas prób symulowania poszczególnych sposobów sterowania modelu wahadła i ich testów.

Rozdział 2

Preliminaria matematyczne

W rozdziale zostaną przedstawione podstawowe metody matematyczne wykorzystane w dalszej części pracy. Pozwoli to na zaznajomienie czytelnika z przyjętą konwencją oznaczeń oraz formalizmami i, w efekcie, na łatwiejsze zrozumienie następujących rozdziałów. Opis zagadnień matematycznych został opracowany na podstawie pozycji literatury [TM18a, TM18b, TM17, SK08, Mur06, LD05, TMD+00, GAK13]. Można w nich znaleźć bardziej szczegółowe opisy poruszanych zagadnień, jak również wprowadzenia oraz dowody.

2.1 Formalizm Lagrange'a

Przyjmując, że ruch układu został opisany za pomocą zmiennych uogólnionych $\mathbf{q} = (q_1, q_2, \dots, q_n)^T$ oraz prędkości uogólnionych $\dot{\mathbf{q}} = (\dot{q}_1, \dot{q}_2, \dots, \dot{q}_n)$, lagranżian układu może zostać określony jako

$$L(\mathbf{q}, \dot{\mathbf{q}}) = K(\mathbf{q}, \dot{\mathbf{q}}) - V(\mathbf{q}), \quad (2.1)$$

gdzie $K(\mathbf{q}, \dot{\mathbf{q}})$ – jego energia kinetyczna układu, natomiast $V(\mathbf{q})$ – energia potencjalna. Wówczas równania ruchu układu mogą zostać otrzymane z równań Eulera-Lagrange'a

$$\frac{d}{dt} \frac{\partial L}{\partial \dot{\mathbf{q}}} - \frac{\partial L}{\partial \mathbf{q}} = \mathbf{F}, \quad (2.2)$$

gdzie \mathbf{F} oznacza siły niepotencjalne działające na układ, takie jak siły sterujące układem, tarcie, opory ruchu. Układ równań (2.2) stanowi układ równań różniczkowych zwyczajnych drugiego rzędu

$$\frac{\partial^2 L(\mathbf{q}, \dot{\mathbf{q}})}{\partial \dot{\mathbf{q}}^2} \ddot{\mathbf{q}} + \frac{\partial^2 L(\mathbf{q}, \dot{\mathbf{q}})}{\partial \mathbf{q} \partial \dot{\mathbf{q}}} \dot{\mathbf{q}} - \frac{\partial L(\mathbf{q}, \dot{\mathbf{q}})}{\partial \mathbf{q}} = \mathbf{F}. \quad (2.3)$$

Zapisanie układu równań dynamiki (2.3) w postaci wektorowej prowadzi do następującej formy

$$\mathbf{Q}(\mathbf{q}) \ddot{\mathbf{q}} + \mathbf{C}(\mathbf{q}, \dot{\mathbf{q}}) \dot{\mathbf{q}} + \mathbf{D}(\mathbf{q}) = \mathbf{F}. \quad (2.4)$$

W powyższym równaniu symetryczna i dodatnie określona macierz $\mathbf{Q}(\mathbf{q})$ jest macierzą bezładności układu, macierz $\mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})$ uwzględnia wpływ sił Coriolisa i odśrodkowych, natomiast wektor $\mathbf{D}(\mathbf{q})$ stanowi wektor sił grawitacji.

2.2 System nieliniowy wejściowo afiniczny

Założmy, że wartość sił niepotencjalnych F z równania (2.4) będzie stanowił sterowanie systemu \mathbf{u} . Przekształcając postać wektorową równań dynamiki (2.4) otrzymujemy

$$\ddot{\mathbf{q}} = -\mathbf{Q}^{-1}(\mathbf{q})(\mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}} + \mathbf{D}(\mathbf{q})) + \mathbf{Q}^{-1}(\mathbf{q})\mathbf{u}. \quad (2.5)$$

Dodając do równań dodatkowe równania prędkości

$$\begin{cases} \dot{\mathbf{q}} = \dot{\mathbf{q}} \\ \ddot{\mathbf{q}} = -\mathbf{Q}^{-1}(\mathbf{q})(\mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}} + \mathbf{D}(\mathbf{q})) + \mathbf{Q}^{-1}(\mathbf{q})\mathbf{u} \end{cases} \quad (2.6)$$

oraz odpowiednio segregując poszczególne elementy, uzyskujemy równanie postaci

$$\dot{\mathbf{x}}(t) = \mathbf{f}(\mathbf{x}(t)) + \mathbf{g}(\mathbf{x}(t))\mathbf{u}(t) = \mathbf{f}(\mathbf{x}(t)) + \sum_{i=1}^m g_i(\mathbf{x}(t))\mathbf{u}_i(t), \quad (2.7)$$

zwanej afiniczną, gdzie wektor stanu $\mathbf{x} = (\mathbf{q} \ \dot{\mathbf{q}})^T$, $\mathbf{f}(\mathbf{x})$ jest dryfem układu, $\mathbf{g}(\mathbf{x})$ jest macierzą pól wektorowych g_1, g_2, \dots, g_m odpowiednio związanych z wartościami u_1, u_2, \dots, u_m sterowań układu. Układ nieliniowy wejściowo afiniczny zachowuje liniową zależność od sterowań układu $\mathbf{u}(t)$.

2.3 Regulator LQR

2.3.1 Równanie stanu i przybliżenie liniowe układu

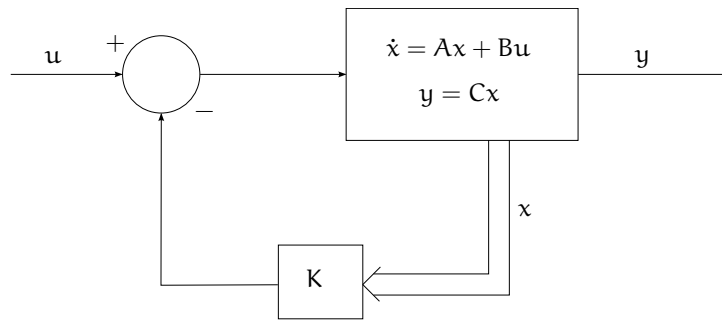
Wiele układów dynamicznych można opisać za pomocą równań stanu

$$\begin{cases} \dot{\mathbf{x}} = \mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{u} \\ \mathbf{y} = \mathbf{C}\mathbf{x} \\ \mathbf{x}(0) = \mathbf{x}_0 \end{cases}, \quad (2.8)$$

gdzie macierz \mathbf{A} jest macierzą stanu, macierz \mathbf{B} macierzą wejścia, macierz \mathbf{C} macierzą wyjścia, \mathbf{x} wektorem stanu systemu, \mathbf{u} sterowaniami, natomiast \mathbf{y} wyjściem układu. Przekształcenie układu nieliniowego danego równanie (2.7) do postaci liniowych równań stanu może zostać dokonane poprzez jego przybliżenie liniowe w danym punkcie pracy. Przybliżenie liniowe można wyznaczyć z pomocą pochodnych cząstkowych policzonych odpowiednio względem kolejnych zmiennych wektora stanu \mathbf{x} oraz wektora sterowania \mathbf{u}

$$\begin{aligned} \mathbf{A} &= \frac{\partial}{\partial \mathbf{x}}(\mathbf{f}(\mathbf{x}) + \mathbf{g}(\mathbf{x})\mathbf{u})|_{\mathbf{x}=\mathbf{x}_0, \mathbf{u}=\mathbf{u}_0}, \\ \mathbf{B} &= \frac{\partial}{\partial \mathbf{u}}(\mathbf{f}(\mathbf{x}) + \mathbf{g}(\mathbf{x})\mathbf{u})|_{\mathbf{x}=\mathbf{x}_0, \mathbf{u}=\mathbf{u}_0}, \end{aligned} \quad (2.9)$$

gdzie \mathbf{x}_0 oraz \mathbf{u}_0 jest punktem pracy, w którym wyznaczamy przybliżenie liniowe układu. Występująca w równaniach (2.8) macierz \mathbf{C} określa, które zmienne wektora stanu \mathbf{x} i w jaki sposób zostaną przekazane na wyjście systemu. Jej postać wynika z własności danego systemu.



Rysunek 2.1 Schemat regulatora LQR

2.3.2 Regulator LQR

Regulator liniowo-kwadratowy (*Linear–quadratic regulator*, LQR) określa rozwiązanie problemu optymalnego sterowania dla systemu dynamicznego opisanego liniowymi równaniami różniczkowymi postaci (2.8) oraz kwadratowym wyznacznikiem jakości. LQR wykorzystuje sprzężenie zwrotne od stanu, jego schemat został przedstawiony na rysunku 2.1.

Celem regulatora LQR jest znalezienie optymalnego prawa sterowania

$$\mathbf{u} = -\mathbf{K}\mathbf{x} \quad (2.10)$$

poprzez minimalizację funkcji kosztu. Dla nieograniczonego horyzontu czasowego funkcja ta przyjmuje formę

$$J = \int_0^{\infty} (\mathbf{X}^T \mathbf{Q} \mathbf{X} + \mathbf{u}^T \mathbf{R} \mathbf{u}) dt, \quad (2.11)$$

gdzie macierz \mathbf{Q} jest symetryczną dodatnio półokreśloną macierzą postaci

$$\mathbf{Q} = \begin{bmatrix} Q_1 & 0 & \dots & 0 \\ 0 & Q_2 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & Q_m \end{bmatrix} \quad (2.12)$$

oraz macierz \mathbf{R} jest macierzą symetryczną, dodatnio określoną postaci

$$\mathbf{R} = \begin{bmatrix} R_1 & 0 & \dots & 0 \\ 0 & R_2 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & R_n \end{bmatrix}. \quad (2.13)$$

Rozmiary macierzy \mathbf{Q} i \mathbf{R} zależą odpowiednio od liczby zmiennych w wektorze stanu \mathbf{x} oraz liczbie sterowań układu \mathbf{u} . Współczynniki Q_1, Q_2, \dots, Q_m określają koszt, jaki układ poniesie za przejście z obecnego stanu układu do stanu układu w punkcie pracy, dla którego jest liczone sterowanie. Współczynniki R_1, R_2, \dots, R_n określają natomiast koszt, jaki układ poniesie za dostarczenie do systemu energii poprzez sterowania \mathbf{u} .

Dla tak określonej funkcji kosztu optymalną macierz sprzężenia od stanu \mathbf{K} , z równania (2.10), można obliczyć zgodnie ze wzorem

$$\mathbf{K} = \mathbf{R}^{-1}(\mathbf{B}^T \mathbf{P}), \quad (2.14)$$

gdzie macierz \mathbf{P} można otrzymać rozwiązując ciągłe równanie Riccatiego

$$\mathbf{A}^T \mathbf{P} + \mathbf{P} \mathbf{A} - (\mathbf{P} \mathbf{B}) \mathbf{R}^{-1} (\mathbf{B}^T \mathbf{P}) + \mathbf{Q} = 0. \quad (2.15)$$

2.3.3 Dobór wartości parametrów macierzy \mathbf{Q} i \mathbf{R}

Poprawne działanie regulatora LQR wymaga odpowiedniego dobrania wartości parametrów w macierzach \mathbf{Q} oraz \mathbf{R} . Do określenia ich początkowych wartości można posłużyć się wzorem

$$q_i = \frac{1}{e_i^2}, \quad (2.16)$$

gdzie e_i jest wartością dopuszczalnego błędu stanu x_i . Na przykład, jeśli rozważamy pojedyncze odwrócone wahadło i chcemy, aby dopuszczalne odchylenie wartości kąta od punktu równowagi chwiejnej, w którym nasz układ ma pozostać, było równa $\frac{1}{30}$ to zgodnie ze wzorem (2.16) wartość parametru q powinna wynosić

$$q = \frac{1}{\left(\frac{1}{30}\right)^2} = 900.$$

Analogicznie można postąpić dla wartości parametrów macierzy \mathbf{R} . Dobór parametrów z wykorzystaniem przytoczonej metody może jednak nie spowodować działania systemu w satysfakcjonujący sposób. Dalsze modyfikacje parametrów mogą odbywać się metodą prób i błędów, na bazie obserwacji zachowania układu po dokonanych zmianach.

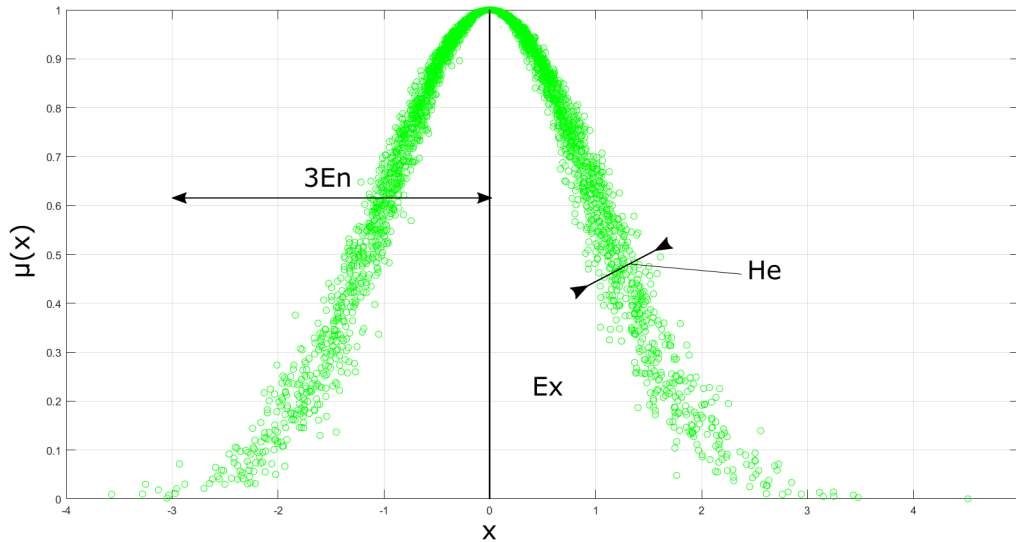
2.4 Model chmury

Model chmury (*Cloud Model*) pozwala na określenie relacji między pojęciami jakościowymi, takimi jak *dobrze*, *źle*, a wartościami ilościowymi, na przykład 10, -10. Zgodnie z definicją podaną w [LD05] założmy, że zbiór \mathbf{U} jest uniwersalnym zbiorem opisanym precyzyjnie za pomocą liczb oraz \mathbf{C} jest jakościowym pojęciem związanym z \mathbf{U} . *Kropką* możemy nazwać każdą wartości $x \in \mathbf{U}$, która losowo realizuje pojęcie \mathbf{C} , natomiast $\mu(x) \in [0, 1]$ stanowi stopień pewności x dla \mathbf{C}

$$\mu : \mathbf{U} \rightarrow [0, 1] \quad \forall x \in \mathbf{U} \quad x \rightarrow \mu(x). \quad (2.17)$$

Chmurą nazywamy rozkład x w \mathbf{U} . Do opisu chmury wykorzystuje się trzy wartości:

- wartość oczekiwana E_x , która jest najbardziej reprezentacyjną wartością dla danego pojęcia jakościowego – można utożsamić ją z wartością oczekiwaną w znaczeniu prawdopodobieństwa,



Rysunek 2.2 Przykładowa chmura wygenerowana dla wartości $Ex = 0$, $En = 1$ oraz $He = 0.1$ (na podstawie [LZG16])

- entropia En , która jest miarą niepewności wartości w danym pojęciu jakościowym – pozwala określić jak bardzo wartość x może być odległa od wartości oczekiwanej Ex , żeby przynależała do danego pojęcia ze stopniem pewności $\mu(x) > 0$,
- hiperentropia He , która jest miarą niepewności entropii – innymi słowy jeżeli wartość He jest równa 0, chmura przyjmuje postać ciągłej linii, a nie kropelk rozrzuconych wokół pewnej linii.

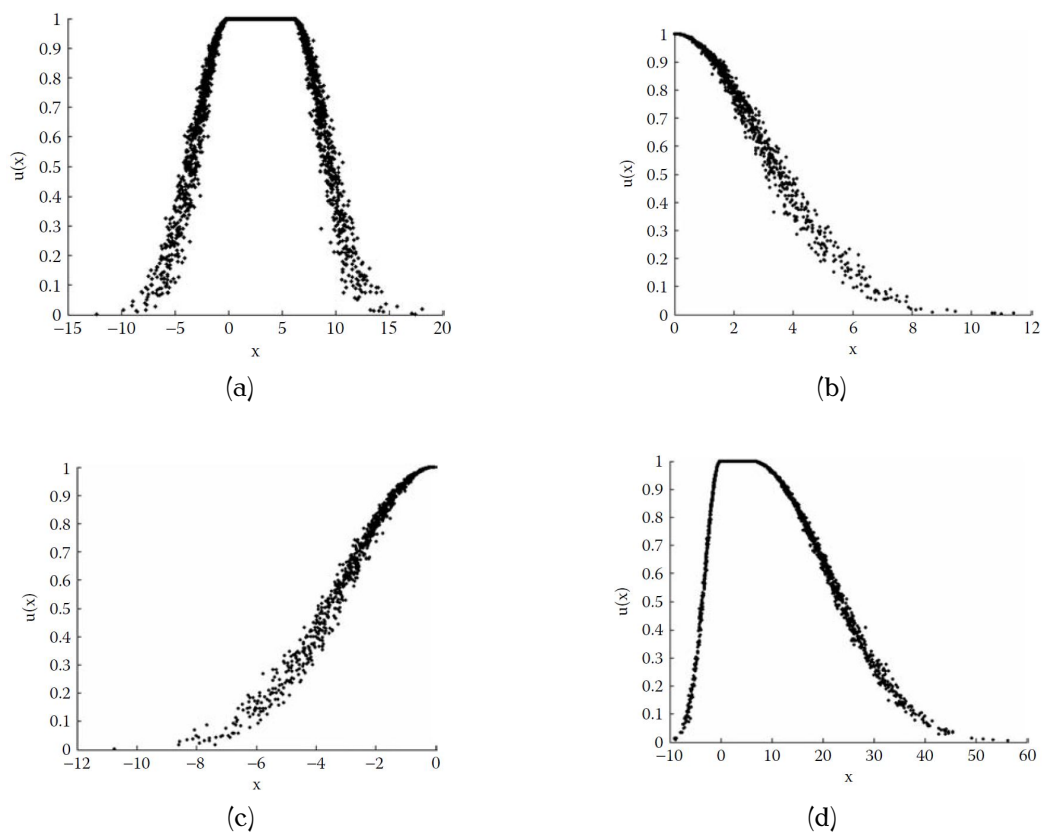
Przykładowa chmura wraz z zaznaczonymi Ex , En , He została przedstawiona na rysunku 2.2.

2.4.1 Rodzaje chmur

Zaprezentowana na rysunku 2.2 chmura przedstawia symetryczny model chmury. Wyróżniamy również modele chmur połowicznych oraz modele chmur asymetrycznych, które są połączone z sobą różnymi chmurami. Wizualizacje różnych typów chmur zostały przedstawione na rysunku 2.3. Pokazana chmura symetryczna jest inną wariacją chmury zaprezentowanej jako przykład.

2.4.2 Generowanie chmury

Zaprezentowane w podrozdziale chmury zostały wykonane na bazie rozkładu normalnego oraz dzwonowej funkcji przynależności. Generowanie chmur odbywa się za pomocą generatorów chmur. Wyróżniamy dwa rodzaje generatorów: *forward normal cloud generator* (CG) oraz *backward normal cloud generator* (CG^{-1}). Pierwszy z nich służy do generowania chmury na podstawie trzech wartości (Ex , EN , He). Schemat generatora został przedstawiony na rysunku 2.4a. Drugi



Rysunek 2.3 Różne rodzaje chmur: a) symetryczna, b), c) połowiczna, d) asymetryczna [LD05]



Rysunek 2.4 Schematy generatorów chmury: a) forward cloud generator, b) backward cloud generator [LD05]

typ generatorów pozwala natomiast odtworzyć wartości (Ex, En, He) na podstawie kropli chmury. Schemat został przedstawiony na rysunku 2.4b. Na potrzeby pracy dokładniej zostanie przybliżona tylko zasada działania generatora CG. Więcej informacji na temat generatora CG^{-1} można znaleźć w [LD05].

Tak jak przy definiowaniu chmury założymy, że zbiór U jest uniwersalnym, ilościowym zbiorem opisanym precyzyjnie przez liczby oraz C jest jakościowym pojęciem związanym ze zbiorem U . Jeżeli istnieje $x \in U$, które realizuje losowo pojęcie C oraz x spełnia $x \sim N(Ex, En'^2)$, gdzie $En' \sim N(En, He^2)$ i stopień pewności x w C jest równy

$$\mu = e^{-\frac{(x-Ex)^2}{2(En')^2}}, \quad (2.18)$$

wówczas rozkład x w U nazywamy chmurą o rozkładzie normalnym (*normal cloud*). Na podstawie definicji zaprezentowanej powyżej można podać prosty algorytm służący do generowania chmur.

Dane wejściowe: wartości (Ex, En, He) oraz liczba kropeł w chmurze n

Dane wyjściowe: n kropeł x z wartościami ich stopnia pewności $\mu(x)$,

$Drop(x_i, \mu_i), i = 1, 2, \dots, n$

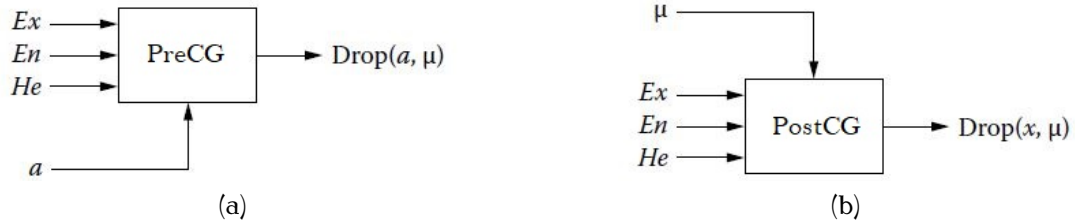
Algorytm

1. Generujemy liczbę En'_i o rozkładzie normalnym z wartością oczekiwaną En oraz wariancją He^2 , $En'_i = N(En, He^2)$.
2. Generujemy liczbę x_i o rozkładzie normalnym z wartością oczekiwaną Ex oraz wariancją En'^2_i , $x_i = N(Ex, En'^2_i)$.
3. Obliczamy

$$\mu_i = e^{-\frac{(x_i-Ex)^2}{2(En'_i)^2}}.$$

4. Łączymy x_i oraz stopień pewności μ_i w jedną krople.
5. Powtarzamy krok od 1 do 4 do momentu wygenerowania n kropeł.

W tym miejscu warto zauważyć, że wartości En oraz He powinny być większe od zera. Jeżeli wartość $He = 0$, wtedy chmura przyjmie postać krzywej dzwonowej, natomiast dla $He = 0$ i $En = 0$, wszystkie wygenerowane wartości x będą dokładnie równe wartości oczekiwanej Ex .



Rysunek 2.5 Schematy generatorów chmury: a) precondition cloud generator, b) postcondition cloud generator (na podstawie [LD05])

2.4.3 Tworzenie reguł opartych o model chmury

Przedstawione w poprzednim podrozdziale model chmury o rozkładzie normalnym mogą zostać wykorzystane do tworzenia zasad opartych o pojęcia jakościowe. Pozwala to na wykonanie bazy reguł w postaci pojęć oraz systemów wnioskowania wykorzystujących tę bazę do sterowania obiektami. Do tworzenia reguł typu

Jeżeli A wtedy B,

gdzie A i B odpowiadają pojęciom C_1 i C_2 odpowiednio w uniwersalnych zbiorach U_1 i U_2 , wymagane będzie wprowadzenie dwóch generatorów. Pierwszy, pokazany na rysunku (2.5a), nazywamy *precondition cloud generator* (PreCG). Pozwala on na obliczenie stopnia pewności μ dla podanej wartości a oraz wartości (Ex, En, He) chmury, która opisuje dane pojęcie C_1 . Drugim generatorem jest *postcondition cloud generator* (PostCG), przedstawiony na rysunku 2.5b. Daje on możliwość określenia wartości x dla podanego stopnia pewności μ oraz wartości (Ex, En, He) chmury, opisującej dane pojęcie C_2 . Nieznacznie modyfikując zaprezentowany w poprzednim podrozdziale algorytm tworzenia chmur (2.4.2), można podać algorytm działania generatorów PreCG oraz PostCG. Algorytm dla generatora *precondition cloud generator* przedstawia się następująco:

Dane wejściowe: wartość a oraz (Ex, En, He) związane z danym pojęciem C

Dane wyjściowe: kropla związaną z wartością a oraz odpowiadający jej stopień niepewności $\mu(a)$

Algorytm

1. Generujemy liczbę En' o rozkładzie normalnym z wartością oczekiwaną Ex oraz wariancją He^2 , $En' = N(Ex, He^2)$.
2. Obliczamy

$$\mu = e^{-\frac{(a-Ex)^2}{2(En')^2}}.$$

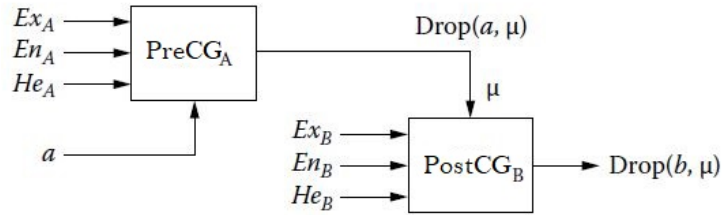
3. Łączymy a oraz wyliczone μ w jedną kroplę.

Analogicznie może zostać zaprezentowany algorytm dla generatora *postcondition cloud generator*:

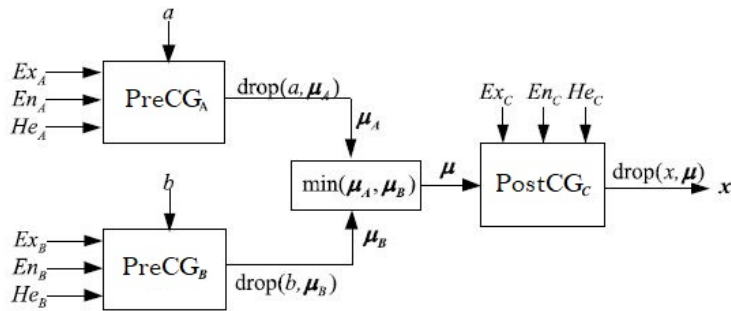
Dane wejściowe: wartość stopnia pewności μ oraz (Ex, En, He) związane z danym pojęciem C

Dane wyjściowe: kropla x związaną ze stopniem niepewności $\mu(a)$

Algorytm



Rysunek 2.6 Generator zasady z pojedynczym warunkiem (na podstawie [LD05])



Rysunek 2.7 Generator zasady z podwójnym warunkiem (na podstawie [LZG16])

1. Generujemy liczbę En' o rozkładzie normalnym z wartością oczekiwaną Ex oraz wariacją He^2 , $En' = N(En, He^2)$.
2. Obliczamy

$$x = Ex \pm En' \sqrt{-2 \ln(\mu)}.$$
3. Łączymy x oraz wyliczone μ w jedną kropkę.

Z zaprezentowanych generatorów można stworzyć jedno warunkową zasadę typu Jeżeli A wtedy B. Schemat odpowiednio połączonych generatorów został przedstawiony na rysunku (2.6). Należy zwrócić uwagę, że generator PostCG generuje dwie wartości x (patrz krok 2 algorytmu). Aby wybrać właściwą należy porównać wartość a z wartością oczekiwaną Ex dla pojęcia A. Jeżeli wartość ta jest mniejsza od wartości oczekiwanej, należy wziąć wartość $x = Ex - En' \sqrt{-2 \ln(\mu)}$. W przeciwnym wypadku wartość $x = Ex + En' \sqrt{-2 \ln(\mu)}$.

Przedstawiony sposób generowania reguł można rozbudowywać na przykład dodając kolejne warunki. Przy podwójnym warunku postać reguły przyjmuje formę

Jeżeli A_1, A_2 wtedy B.

Schemat połączonych generatorów został przedstawiony na rysunku 2.7. Można zauważyć, że dodanie kolejnego warunku powoduje dodanie kolejnego generatora PreCG. Następnie stopnie pewności μ_a i μ_b dla podanych wartości a oraz b są ze sobą porównywane i wybierana jest najmniejsza wartość stopnia pewności, która jest w dalszym etapie wejściem do generatora PostCG i na jej podstawie otrzymujemy końcową wartość x .

2.5 Generowanie trajektorii ruchu

Projektowanie trajektorii ruchu odwróconego, podwójnego wahadła będzie bazowało na postaci afinicznej równań ruchu (2.7). Z postaci tej zostały wyodrębnione równania związane z przyspieszeniami i zapisane jako układ równań

$$\begin{cases} \ddot{s} = u \\ \ddot{\phi} = F(\phi, \dot{\phi}) + G(\phi)u \end{cases} \quad (2.19)$$

Przy projektowaniu trajektorii wykorzystamy funkcje nasycenia związane z wartością przemieszczenia wózka s , prędkością wózka \dot{s} oraz przyspieszeniem wózka \ddot{s} [GAK13]. Pozwoli to na wyliczenie trajektorii w taki sposób, aby była dopasowana do fizycznego modelu wahadła, na przykład długości szyny, po której będzie poruszać się wózek czy przyspieszenia, jakie można wygenerować za pomocą wybranego silnika. Wymagane jest, aby funkcje nasycenia dla s oraz \dot{s} były różniczkowalne. Z tego powodu zostały wybrane funkcje dane równaniem

$$\psi_i(\xi_i, \psi_i^\pm) = \psi_i^+ - \frac{\psi_i^+ - \psi_i^-}{1 + \exp\left(\frac{4\xi_i}{\psi_i^+ - \psi_i^-}\right)} \quad (2.20)$$

dla $i = 1, 2$, gdzie ξ_i są nowo wprowadzonymi zmiennymi stanu i ψ_i^\pm są górnym i dolnym ograniczeniem funkcji nasycenia. Trzecia funkcja nasycenia nie musi być różniczkowalna, jednakże powinna dać możliwość zamodelowania odpowiedniego, cyklicznego ruchu wózka. Do tego celu została użyta funkcja

$$\psi_3(v, \psi_3^\pm) = \begin{cases} \psi_3^+ & \text{dla } v > \psi_3^+ \\ v & \text{dla } v \in [\psi_3^-, \psi_3^+] \\ \psi_3^- & \text{dla } v < \psi_3^- \end{cases}, \quad (2.21)$$

gdzie ψ_3^\pm są ponownie górnym i dolnym ograniczeniem, a v jest funkcją modelującą zachowanie wózka pozwalającą zrealizować zaplanowaną trajektorię. Jej wartości można obliczyć za pomocą równania

$$v = \Phi(t, \mathbf{p}) = \sum_{i=1}^6 p_i \sin\left(\frac{h_i \pi t}{T}\right), \quad (2.22)$$

gdzie h_i są dodatnimi, kolejnymi liczbami naturalnymi spełniającymi warunek $h_i \neq h_j, \forall i \neq j$, T jest czasem wykonywania manewru przez wahadło, natomiast p_i są współczynnikami pozwalającymi odpowiednio dopasować funkcje do manewru. Wykorzystując tak zdefiniowane funkcje nasycenia można zapisać, że

$$s = \psi_1(\xi_1, \psi_1^\pm). \quad (2.23)$$

Następnie, różniczkując wyrażenie dwukrotnie otrzymujemy

$$\dot{s} = \frac{\partial \psi_1}{\partial \xi_1} \dot{\xi}_1 \quad (2.24)$$

oraz

$$\ddot{s} = \frac{\partial^2 \psi_1}{\partial \xi_1^2} \dot{\xi}_1^2 + \frac{\partial \psi_1}{\partial \xi_1} \ddot{\xi}_1, \quad (2.25)$$

gdzie

$$\dot{\xi}_1 = \psi_2(\xi_2, \psi_2^\pm(\xi_1)), \quad (2.26)$$

a także

$$\dot{\xi}_2 = \psi_3(v, \psi_3^\pm(\xi_1, \xi_2)). \quad (2.27)$$

Założmy, że wartości s , \dot{s} i \ddot{s} należą kolejno do przedziałów $[s_0^-, s_0^+]$, $[s_1^-, s_1^+]$, $[s_2^-, s_2^+]$ oraz $s_i^+ = -s_i^-$ dla $i = 0, 1, 2$. Ograniczenia

$$\psi_1^\pm = s_0^\pm. \quad (2.28)$$

Zgodnie z równaniami (2.24) i (2.26) możemy zapisać

$$s_1^- \leq \frac{\partial \psi_1}{\partial \xi_1} \psi_2 \leq s_1^+. \quad (2.29)$$

Przekształcając tą nierówność otrzymujemy wartość ograniczeń

$$\psi_2^\pm(\xi_1) = s_1^\pm \left[\frac{\partial \psi_1}{\partial \xi_1} \right]^{-1}. \quad (2.30)$$

Analogicznie można wyliczyć ograniczenia

$$\psi_3^\pm(\xi_1, \xi_2) = s_2^\pm \left[\frac{\partial \psi_1}{\partial \xi_1} \frac{\partial \psi_2}{\partial \xi_2} \right]^{-1} - \left[\frac{\partial^2 \psi_1}{\partial \xi_1^2} \psi_2^2 + \frac{\partial \psi_1}{\partial \xi_1} \frac{\partial \psi_2}{\xi_1} \psi_2 \right] \left[\frac{\partial \psi_1}{\partial \xi_1} \frac{\partial \psi_2}{\partial \xi_2} \right]^{-1}. \quad (2.31)$$

Zapisując układ równań (2.19) z wykorzystaniem wprowadzonych zmiennych ξ_1 i ξ_2 otrzymujemy

$$\begin{cases} \dot{\xi}_1 = \psi_2(\xi_2, \psi_2^\pm(\xi_1)) \\ \dot{\xi}_2 = \psi_3(v, \psi_3^\pm(\xi_1, \xi_2)) \\ \ddot{\boldsymbol{\varphi}} = \mathbf{F}(\boldsymbol{\varphi}, \dot{\boldsymbol{\varphi}}) + \mathbf{G}(\boldsymbol{\varphi})\mathbf{u} \end{cases}, \quad (2.32)$$

gdzie zgodnie z równaniem (2.25)

$$\mathbf{u} = \frac{\partial^2 \psi_1}{\partial \xi_1^2} \psi_2^2 + \frac{\partial \psi_1}{\partial \xi_1} \left[\frac{\partial \psi_2}{\partial \xi_1} \psi_2 + \frac{\partial \psi_2}{\partial \xi_2} \psi_3(\Phi(t, \mathbf{p}), \psi_3^\pm(\xi_1, \xi_2)) \right] \quad (2.33)$$

Aby wygenerować trajektorie, należy znaleźć rozwiązanie układu równań (2.32) przy odpowiednio dobranych warunkach brzegowych i zadanych ograniczeniach. I tak, do wygenerowania przykładowo trajektorii między pozycją wahadła pionowo w dół, a pionowo w górę, należy rozwiązać układ (2.32) przy warunkach brzegowych $s(0) = s_0 = 0$, $\boldsymbol{\varphi}^T(0) = \boldsymbol{\varphi}_0 = (\pi, \pi)^T$, $\dot{s}(0) = 0$, $\dot{\boldsymbol{\varphi}}(0) = \dot{\boldsymbol{\varphi}}_0 = \mathbf{0}$, $s(T) = s_T = 0$, $\boldsymbol{\varphi}(T) = \boldsymbol{\varphi}_T = (0, 0)^T$, $\dot{s}(T) = 0$, $\dot{\boldsymbol{\varphi}}(T) = \dot{\boldsymbol{\varphi}}_T = \mathbf{0}$, $\xi_{1,0} = \psi_1^{-1}(s_0, \psi_1^\pm)$, $\xi_{1,T} = \psi_1^{-1}(s_T, \psi_1^\pm)$, $\xi_{2,0} = 0$, $\xi_{2,T} = 0$ oraz ograniczeniach (2.28), (2.30) i (2.31).

Rozdział 3

Podwójne odwrócone wahadło

W rozdziale zostaną przedstawione modele matematyczne podwójnego, odwróconego wahadła. Równania ruchu układu zostaną wyprowadzone z wykorzystaniem formalizmu Lagrange'a. Równania zostaną zapisane w postaci wektorowej, a następnie przekształcone do postaci afinicznego systemu nieliniowego.

Przy wyprowadzeniu modelu założono, że wahadło składa się z dwóch nieważkich prętów połączonych przegubem z punktowymi masami umieszczonymi w ich połowie i jest zamocowana wahlwie na przemieszczającym się w poziomie, napędzanym wózku (zobacz rysunek 3.1). Rozdział zawiera dwa modele: nieuwzględniający i uwzględniający tarcie wiskotyczne w przegubach wahadła.

3.1 Wyprowadzenie równań ruchu

Jako współrzędne uogólnione, opisujące ruch układu, zostały wybrane kąty φ_1 i φ_2 między ramionami wahadła a osią pionową oraz przesunięcie wózka s , względem początku układu współrzędnych. Współrzędne te stanowią wektor konfiguracyjny w postaci

$$\mathbf{q} = \begin{pmatrix} \varphi_1 \\ \varphi_2 \\ s \end{pmatrix} \quad (3.1)$$

Zgodnie z przyjętym wektorem konfiguracji \mathbf{q} , współrzędne punktów środka masy można opisać jako

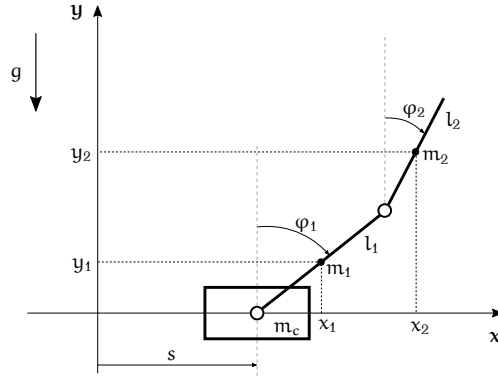
$$\mathbf{p}_1 = \begin{pmatrix} x_1 \\ y_1 \end{pmatrix} = \begin{pmatrix} s - \frac{l_1}{2} \sin(\varphi_1) \\ \frac{l_1}{2} \cos(\varphi_1) \end{pmatrix} \quad (3.2)$$

oraz

$$\mathbf{p}_2 = \begin{pmatrix} x_2 \\ y_2 \end{pmatrix} = \begin{pmatrix} s - l_1 \sin(\varphi_1) - \frac{l_2}{2} \sin(\varphi_2) \\ l_1 \cos(\varphi_1) + \frac{l_2}{2} \cos(\varphi_2) \end{pmatrix}. \quad (3.3)$$

W celu wyznaczenia modelu wahadła z wykorzystaniem zaprezentowanego w podrozdziale 2.1 formalizmu Lagrange'a należy wyznaczyć jego energię kinetyczną i potencjalną. Energia kinetyczna wahadła wynosi

$$K(\mathbf{q}, \dot{\mathbf{q}}) = \frac{1}{2} m_c \dot{s}^2 + \frac{1}{2} \sum_{j=1}^2 m_j \dot{\mathbf{p}}_{cj}^T \dot{\mathbf{p}}_{cj}, \quad (3.4)$$



Rysunek 3.1 Schemat odwróconego, podwójnego wahadła

natomiast energia potencjalna

$$V(\mathbf{q}) = g(m_1 y_1 + m_2 y_2). \quad (3.5)$$

Wówczas lagranżian (2.1) jest dany jako

$$L(\mathbf{q}, \dot{\mathbf{q}}) = \frac{1}{2} m_c \dot{s}^2 + \frac{1}{2} \sum_{j=1}^2 m_j \dot{\mathbf{p}}_{cj}^T \dot{\mathbf{p}}_{cj} - g(m_1 y_1 + m_2 y_2). \quad (3.6)$$

Przyjmując, że jedyną niepotencjalną siłą działającą na układ jest powodująca przemieszczenie wózka siła τ . Stąd występujący w równaniach (2.2) wektor siły \mathbf{F} przyjmuje postać

$$\mathbf{F} = \begin{pmatrix} 0 \\ 0 \\ \tau \end{pmatrix}. \quad (3.7)$$

Wprowadzona siła τ będzie stanowiło sterowanie układu. Wyliczając równanie (2.2) otrzymujemy model dynamiki analizowanego wahadła w postaci wektorowej (2.4) z macierzą bezwładności $\mathbf{Q}(\mathbf{q})$ równą

$$\mathbf{Q}(\mathbf{q}) = \begin{bmatrix} l_1^2(\frac{1}{2}m_1 + m_2) & \frac{1}{2}l_2 l_1 m_2 \cos(\varphi_1 - \varphi_2) & -l_1(\frac{1}{2}m_1 + m_2)\cos(\varphi_1) \\ \frac{1}{2}l_2 l_1 m_2 \cos(\varphi_1 - \varphi_2) & \frac{1}{4}l_2^2 m_2 & -\frac{1}{2}l_2 m_2 \cos(\varphi_2) \\ -l_1(\frac{1}{2}m_1 + m_2)\cos(\varphi_1) & -\frac{1}{2}l_2 m_2 \cos(\varphi_2) & m_1 + m_2 + m_c \end{bmatrix}, \quad (3.8)$$

macierzą sił Coriolisa oraz odśrodkowych

$$\mathbf{C}(\mathbf{q}, \dot{\mathbf{q}}) = \begin{bmatrix} 0 & \frac{1}{2}l_2 l_1 m_2 \sin(\varphi_1 - \varphi_2) \dot{\varphi}_2 & 0 \\ -\frac{1}{2}l_2 l_1 m_2 \sin(\varphi_1 - \varphi_2) \dot{\varphi}_2 & 0 & 0 \\ l_1(\frac{1}{2}m_1 + m_2)\sin(\varphi_1) \dot{\varphi}_1 & \frac{1}{2}l_2 m_2 \sin(\varphi_2) \dot{\varphi}_2 & 0 \end{bmatrix} \quad (3.9)$$

oraz wektorem sił grawitacji

$$\mathbf{D}(\mathbf{q}) = \begin{bmatrix} -gl_1(\frac{1}{2}m_1 + m_2)\sin(\varphi_1) \\ -\frac{1}{2}gl_2 m_2 \sin(\varphi_2) \\ 0 \end{bmatrix}. \quad (3.10)$$

3.2 Przekształcenie do postaci afinicznej

Po otrzymaniu równań dynamiki podwójnego, odwróconego wahadła w formie wektorowej, przekształcimy je do postaci wejściowo afinicznego systemu nieliniowego. Zgodnie z równaniem (2.7) system składa się z dryfu oraz części związanej ze sterowaniami. Z uwagi na fakt, że odwrócone wahadło jest sterowane jedynie za pomocą siły zewnętrznej τ przyłożonej do wózka, która bezpośrednio wpływa na jego przyspieszenie \ddot{s} , sterowanie $u(t)$ układu (2.7) jest skalarem w postaci rzeczownego przyspieszenia \ddot{s} . Wydzielając z macierzy $\mathbf{Q}(\mathbf{q})$, $\mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})$ oraz wektora $\mathbf{D}(\mathbf{q})$, danych równaniami (3.8) – (3.10), część związaną z przyspieszeniem \ddot{s} , otrzymujemy

$$\mathbf{Q}_{\varphi\varphi} = \begin{bmatrix} l_1^2(\frac{m_1}{4} + m_2) & \frac{1}{2}l_2l_1m_2\cos(\varphi_1 - \varphi_2) \\ \frac{1}{2}l_2l_1m_2\cos(\varphi_1 - \varphi_2) & \frac{1}{4}m_2l_2^2 \end{bmatrix}, \quad (3.11)$$

$$\mathbf{Q}_{\varphi s} = \begin{bmatrix} -l_1(\frac{1}{2}m_1 + m_2)\cos(\varphi_1) \\ -\frac{1}{2}l_2m_2\cos(\varphi_2) \end{bmatrix}, \quad (3.12)$$

$$\mathbf{C}_{\varphi\varphi}(\boldsymbol{\varphi}, \dot{\boldsymbol{\varphi}}) = \begin{bmatrix} 0 & \frac{1}{2}l_2l_1m_2\sin(\varphi_1 - \varphi_2)\dot{\varphi}_2 \\ -\frac{1}{2}l_2l_1m_2\sin(\varphi_1 - \varphi_2)\dot{\varphi}_2 & 0 \end{bmatrix}, \quad (3.13)$$

$$\mathbf{D}_{\varphi\varphi}(\boldsymbol{\varphi}) = \begin{bmatrix} -gl_1(\frac{1}{2}m_1 + m_2)\sin(\varphi_1) \\ -\frac{1}{2}gl_2m_2\sin(\varphi_2) \end{bmatrix}. \quad (3.14)$$

Porządkując tak otrzymane równanie uzyskujemy

$$\mathbf{Q}_{\varphi\varphi}(\boldsymbol{\varphi})\ddot{\boldsymbol{\varphi}} = -\mathbf{Q}_{\varphi s}(\boldsymbol{\varphi})\ddot{s} - \mathbf{C}_{\varphi\varphi}(\boldsymbol{\varphi}, \dot{\boldsymbol{\varphi}})\dot{\boldsymbol{\varphi}} - \mathbf{D}_{\varphi\varphi}(\boldsymbol{\varphi}) \quad (3.15)$$

z $\boldsymbol{\varphi} = (\varphi_1, \varphi_2)^\top$. Na podstawie tak przekształconych równań dynamiki można zapisać postać afiniczną modelu podwójnego, odwróconego wahadła jako

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}) + \mathbf{g}(\mathbf{x})u \quad (3.16)$$

gdzie $\mathbf{x} = (\mathbf{q}, \dot{\mathbf{q}})^\top$, $\mathbf{q} = (\varphi_1, \varphi_2, s)^\top$, $\mathbf{f}(\mathbf{x}) = (\dot{\mathbf{q}}, \mathbf{F}(\boldsymbol{\varphi}, \dot{\boldsymbol{\varphi}}), 0)^\top$ i $\mathbf{g}(\mathbf{x}) = (\mathbf{0}_{1 \times 3}, \mathbf{G}(\boldsymbol{\varphi}), 1)^\top$ oraz $\mathbf{F}(\boldsymbol{\varphi}, \dot{\boldsymbol{\varphi}}) = -\mathbf{Q}_{\varphi\varphi}^{-1}(\boldsymbol{\varphi})(\mathbf{C}_{\varphi\varphi}(\boldsymbol{\varphi}, \dot{\boldsymbol{\varphi}})\dot{\boldsymbol{\varphi}} + \mathbf{D}_{\varphi\varphi}(\boldsymbol{\varphi}))$, $\mathbf{G}(\boldsymbol{\varphi}) = -\mathbf{Q}_{\varphi\varphi}^{-1}(\boldsymbol{\varphi})\mathbf{Q}_{\varphi s}(\boldsymbol{\varphi})$.

3.3 Uwzględnienie tarcia

Przedstawione w poprzednich podrozdziałach modele matematyczne podwójnego, odwróconego wahadła nie uwzględniają występowania tarcia w przegubach. Do jego zamodelowania przyjmujemy model tarcia wiskotycznego opisany funkcją rozpraszania Rayleigha postaci [GAK13]

$$R = \frac{1}{2}d_1\dot{\varphi}_1^2 + \frac{1}{2}d_2(\dot{\varphi}_2 - \dot{\varphi}_1)^2, \quad (3.17)$$

gdzie d_1 i d_2 są współczynnikami tarcia wiskotycznego występującego odpowiednio w pierwszym oraz drugim przegubie. Równania Eulera-Lagrange'a (2.2) z uwzględnieniem tarcia dla zaprezentowanego wahadła dane jako

$$\frac{d}{dt} \frac{\partial L}{\partial \dot{\mathbf{q}}} - \frac{\partial L}{\partial \mathbf{q}} + \frac{\partial}{\partial \dot{\mathbf{q}}} R = \mathbf{F}, \quad (3.18)$$

co zapisane w postaci wektorowej daje

$$\mathbf{Q}(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}} + \mathbf{G}\dot{\mathbf{q}} + \mathbf{D}(\mathbf{q}) = \mathbf{F}, \quad (3.19)$$

gdzie macierze $\mathbf{Q}(\mathbf{q})$, $\mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})$, wektor $\mathbf{D}(\mathbf{q})$ oraz siła \mathbf{F} są takie jak zdefiniowane w podrozdziale 3.1, natomiast macierz \mathbf{G} jest równa

$$\mathbf{G} = \begin{bmatrix} d_1 + d_2 & -d_2 & 0 \\ -d_2 & d_2 & 0 \\ 0 & 0 & 0 \end{bmatrix}. \quad (3.20)$$

W sposób analogiczny do przedstawionego w podrozdziale 3.2, równanie (3.19) można zapisać w postaci afinicznej

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}) + \mathbf{g}(\mathbf{x})u, \quad (3.21)$$

gdzie $\mathbf{x} = (\mathbf{q}, \dot{\mathbf{q}})^T$, $\mathbf{q} = (\varphi_1, \varphi_2, s)^T$, $\mathbf{f}(\mathbf{x}) = (\dot{\mathbf{q}}, \mathbf{F}(\boldsymbol{\varphi}, \dot{\boldsymbol{\varphi}}), 0)^T$ i $\mathbf{g}(\mathbf{x}) = (\mathbf{0}_{1 \times 3}, \mathbf{G}(\boldsymbol{\varphi}), 1)^T$ oraz $\mathbf{F}(\boldsymbol{\varphi}, \dot{\boldsymbol{\varphi}}) = -\mathbf{Q}_{\varphi\varphi}^{-1}(\boldsymbol{\varphi})(\mathbf{C}_{\varphi\varphi}(\boldsymbol{\varphi}, \dot{\boldsymbol{\varphi}})\dot{\boldsymbol{\varphi}} + \mathbf{G}_{\varphi\varphi}\dot{\boldsymbol{\varphi}} + \mathbf{D}_{\varphi\varphi}(\boldsymbol{\varphi}))$, $\mathbf{G}(\boldsymbol{\varphi}) = -\mathbf{Q}_{\varphi\varphi}^{-1}(\boldsymbol{\varphi})\mathbf{Q}_{\varphi s}(\boldsymbol{\varphi})$

i $\mathbf{G}_{\varphi\varphi} = \begin{bmatrix} d_1 + d_2 & -d_2 \\ -d_2 & d_2 \end{bmatrix}$.

Rozdział 4

Sterowanie do punktu

W rozdziale zostaną przedstawione algorytmy pozwalające na sterowanie podwójnym, odwróconym wahadłem do punktu. W zaprezentowanym przykładzie, zadaniem jakie ma zostać zrealizowane przez regulator będzie próba ustawienia oraz utrzymania wahadła w punkcie równowagi chwiejnej ($\varphi_1 = 0$, $\varphi_2 = 0$). Na początku rozdziału zostanie opisany sposób wykorzystania regulatora LQR do tak postawionego zadania. Następnie zostanie dokonana modyfikacja układu sterowania poprzez dodanie regulatora PID. Na koniec przedstawimy algorytm modyfikowania nastaw regulatora PID wykorzystujący model chmury opisany w podrozdziale 2.4.

4.1 Regulator LQR

Opis zasady działania regulatora LQR został dokładniej przedstawiony w podrozdziale 2.3. Poniżej zaprezentujemy algorytm pozwalający na implementację regulatora LQR jako układu sterowania podwójnym, odwróconym wahadłem.

Pierwszym krokiem, jaki należy wykonać by zastosować regulator LQR, jest otrzymanie równania stanu poprzez przybliżenie liniowe równań dynamiki w wybranym punkcie pracy. Zgodnie z metodologią zaprezentowaną w podrozdziale 2.3.1, można tego dokonać poprzez wyliczenie odpowiednio pochodnych cząstkowych równań w postaci afinicznej (2.7) względem współrzędnych wektora stanu \mathbf{x} oraz sterowań \mathbf{u} z równań dynamiki w postaci afinicznej. Alternatywnie można wyliczyć przybliżenie liniowe równań dynamiki (2.4) zapisanych w postaci wektorowej. Wymaga to jednak najpierw przekształcenia do postaci

$$\ddot{\mathbf{q}} = \mathbf{Q}^{-1}(\mathbf{q})\mathbf{F} - \mathbf{Q}^{-1}(\mathbf{q})(\mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}} + \mathbf{D}(\mathbf{q})), \quad (4.1)$$

a następnie policzenia w analogiczny sposób pochodnych cząstkowych prawej strony tego równania, z przyjętym wektorem sterowań $\mathbf{u} = \mathbf{F}$.

Kolejnym krokiem jest określenie macierzy \mathbf{Q} (2.12) oraz \mathbf{R} (2.13). Dobór elementów macierzy może odbyć się zgodnie z zasadą podaną w podrozdziale 2.3.3. Należy jednak pamiętać, że zaprezentowana tam metoda pozwala jedynie wstępnie określić wartości elementów składowych macierzy \mathbf{Q} oraz \mathbf{R} .

Następnym krokiem jest wyliczenie macierzy \mathbf{P} z wykorzystaniem ciągłego równania Riccatiego (2.15).

Tabela 4.1 Parametry nominalne modelu podwójnego, odwróconego wahadła

Parametr	Wartość
m_c	1[kg]
m_1	0.13[kg]
m_2	0.522[kg]
l_1	0.0775[m]
l_2	0.25[m]

Ostatni krok, jaki należy podjąć, prowadzi do wyliczenia macierzy sprzężenia zwrotnego \mathbf{K} na podstawie wzoru (2.14). W rezultacie otrzymujemy regułę sterowania postaci

$$\mathbf{u} = -\mathbf{K}\mathbf{x}. \quad (4.2)$$

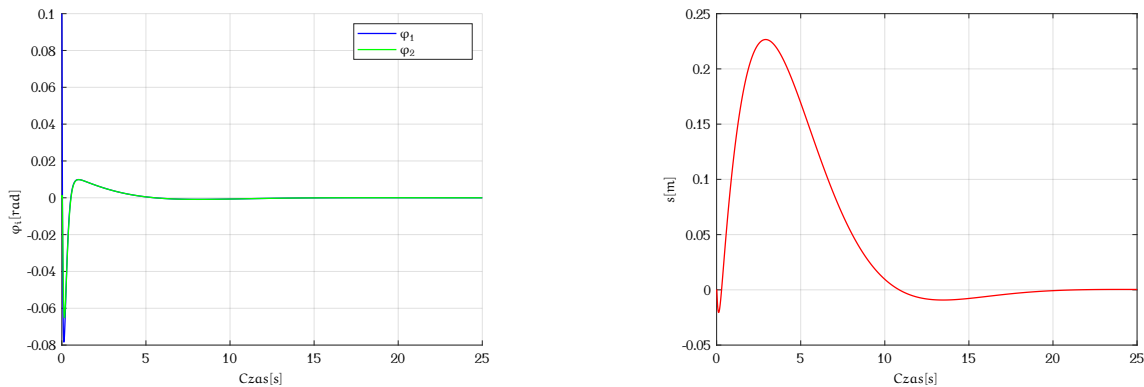
4.1.1 Przykład zastosowania regulatora LQR

Regulator LQR został zaimplementowany dla podwójnego, odwróconego wahadła zgodnie z podanym powyżej algorytmem. Równania dynamiki w postaci afinicznej dla wahadła zostały podane w podrozdziale 3.2. Na ich podstawie, z wykorzystaniem systemu Mathematica, zostały wyliczone macierze \mathbf{A} oraz \mathbf{B} . Wektor stanu, dla którego zostało wyliczone przybliżenie liniowe, był równy $\mathbf{x} = (0, 0, 0, 0, 0, 0)^T$ oraz sterowanie $\mathbf{u} = 0$. Kolejno przyjmując wartości dopuszczalnych błędów $e_i = \frac{1}{10}$ dla $i = 1, \dots, 4$, $e_5 = \frac{1}{22}$ i $e_6 = \frac{10}{45}$, zostały dobrane na podstawie formuły (2.16) wartości macierzy

$$\mathbf{Q} = \begin{bmatrix} 100 & 0 & 0 & 0 & 0 & 0 \\ 0 & 100 & 0 & 0 & 0 & 0 \\ 0 & 0 & 100 & 0 & 0 & 0 \\ 0 & 0 & 0 & 100 & 0 & 0 \\ 0 & 0 & 0 & 0 & 500 & 0 \\ 0 & 0 & 0 & 0 & 0 & 20 \end{bmatrix} * \quad (4.3)$$

oraz, z uwagi na skalarne sterowanie \mathbf{u} , macierz \mathbf{R} w postaci pojedynczej wartości $R = 1000$. Następnie, na podstawie uzyskanych macierzy, za pomocą funkcji *lqr* dostępnej w programie Matlab, policzono macierz sprzężenia zwrotnego $\mathbf{K} = (-327.22, 330.15, -0.32, -3.39, 8.45, -1.15)$. Symulacje działania systemu z regulatorem LQR zostały przeprowadzone z wykorzystaniem pakietu Simulink dostarczanego z oprogramowaniem Matlab. Wartości mas oraz długości poszczególnych części wahadła, zgodnie z oznaczeniami na rysunkach 3.1 zostały podane w tabeli 4.1. Wykresy na rysunku 4.1 przedstawiają przebieg kątów φ_1 i φ_2 oraz przesunięcia wózka s dla wahadła sterowanego regulatorem LQR. Początkowa wartość wektora stanu przyjęto jako $\mathbf{x}_0 = (0.1, 0, 0, 0, 0, 0)^T$. Jak widać, w analizowanej sytu-

*Ustawienie dużej wartości w macierzy \mathbf{Q} dla prędkości kątowej na drugim przegubie ma zapobiec łamaniu się wahadła podczas jego stabilizacji w punkcie równowagi chwiejnej. Pozwala to na łatwiejsze jegoysterowanie. Natomiast mała wartość dla prędkości wózka jest spowodowana faktem, że nie zależy nam na utrzymaniu małych przeregulowań dla tej składowej wektora $\dot{\mathbf{q}}$. Takie ustawienie pozwala również na łatwiejsze sterowanie obiektem, ponieważ regulator może używać większych prędkości podczas ustawiania wahadła w punkcie równowagi chwiejnej.



Rysunek 4.1 Przykładowy przebieg wektora konfiguracji $\mathbf{q} = (\varphi_1, \varphi_2, s)^T$ uzyskany z zastosowaniem regulatora LQR

Tabela 4.2 Alternatywne wartości parametrów modelu podwójnego, odwróconego wahadła

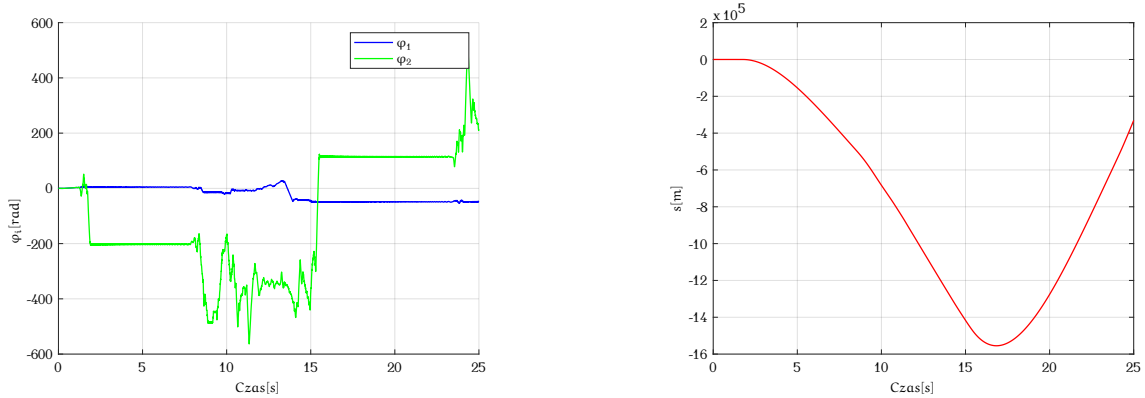
Parametr	Wartość
m_c	1[kg]
m_1	0.5[kg]
m_2	10[kg]
l_1	3[m]
l_2	1.3[m]

acji regulator LQR poprawnie stabilizuje wahadło w punkcie równowagi chwiejnej $\mathbf{q} = (0, 0, 0)^T$ przy małym, początkowym jego odchyleniu.

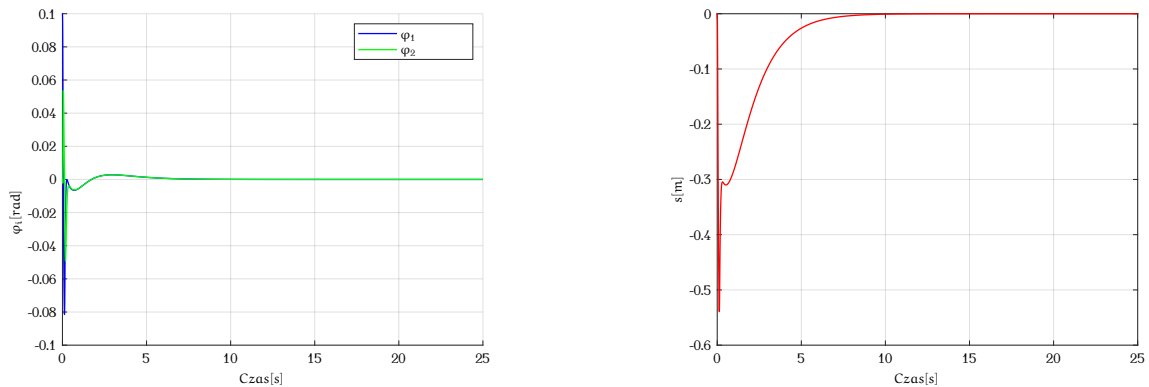
4.1.2 Badanie właściwości regulatora LQR

Pierwszy test ma na celu pokazanie, że wartości macierzy \mathbf{Q} oraz \mathbf{R} należy dobrać w zależności od parametrów wahadła. Alternatywne wartości parametrów wahadła zostały przedstawione w tabeli 4.2. Dla tak określonego wahadła przeprowadzono symulacje z przyjętymi identycznymi wartościami macierzy \mathbf{Q} , \mathbf{R} oraz początkową konfiguracją jak w przykładzie z podrozdziału 4.1.1. Na rysunku 4.2 przedstawiono przebieg kątów oraz pozycję wózka. Wynika z nich, że regulator nie poradził sobie z zadaniem. Dobranie niewłaściwych parametrów macierzy \mathbf{Q} oraz \mathbf{R} może prowadzić do niepoprawnego działania układu sterowania, mimo że dla podobnego obiektu sterowanie jest poprawne.

W drugim teście pokażemy, że zmiana macierzy \mathbf{Q} oraz \mathbf{R} pozwala na poprawne działanie regulatora LQR dla rozpatrywanego tu wahadła. Macierz \mathbf{Q} przyjmuje



Rysunek 4.2 Przebieg składowych wektora q przy alternatywnych wartościach parametrów wahadła i niezmienionych wartościach regulatora LQR



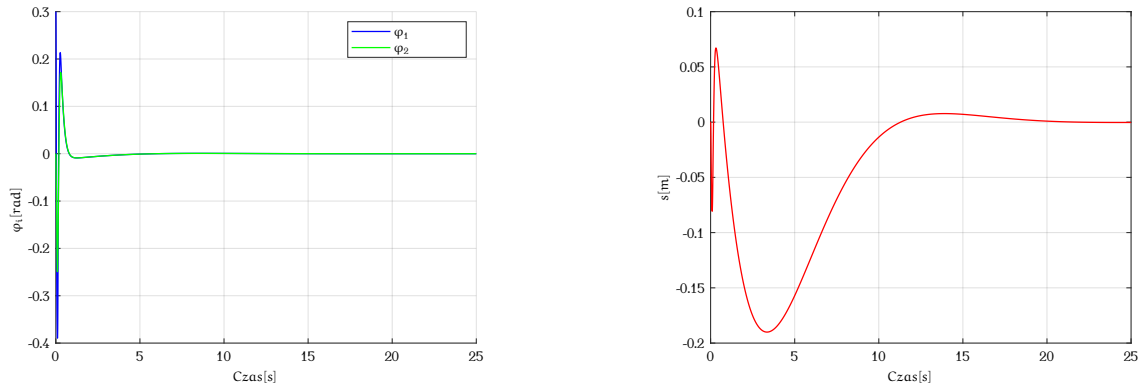
Rysunek 4.3 Przebieg składowych wektora q przy alternatywnych wartościach parametrów wahadła i zmienionych wartościach regulatora LQR

teraz postać

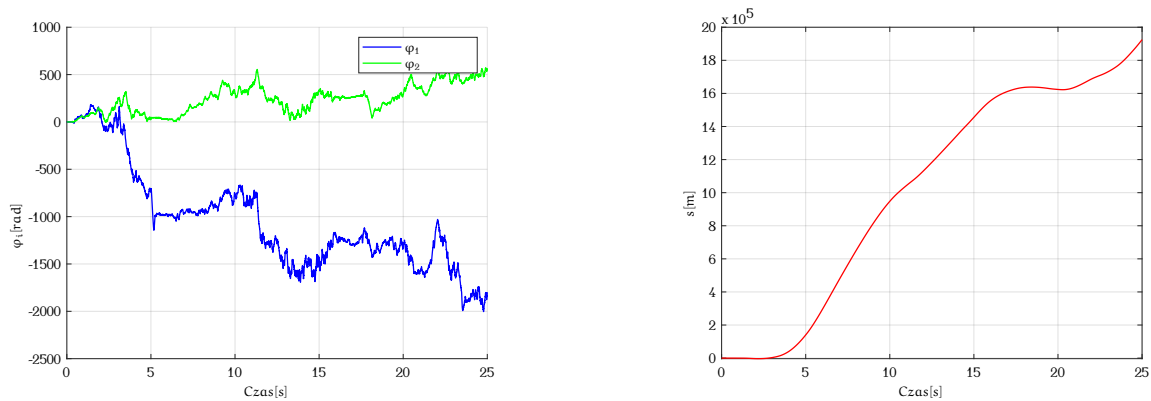
$$Q = \begin{bmatrix} 9000 & 0 & 0 & 0 & 0 & 0 \\ 0 & 100 & 0 & 0 & 0 & 0 \\ 0 & 0 & 9000 & 0 & 0 & 0 \\ 0 & 0 & 0 & 100 & 0 & 0 \\ 0 & 0 & 0 & 0 & 5000 & 0 \\ 0 & 0 & 0 & 0 & 0 & 20000 \end{bmatrix}, \quad (4.4)$$

natomiast $R = 10$. Wykresy ilustrujące działanie regulatora w takim przypadku zostały zaprezentowane na rysunku 4.3.

Kolejne testy będą dotyczyć wpływu początkowej konfiguracji wahadła na skuteczność regulatora LQR. Zostanie pokazane, że regulator radzi sobie tylko z pewnymi małymi odchyleniami od punktu równowagi chwiejnej. Na początek dla wartości parametrów wahadła przedstawionych w tabeli 4.1 oraz wartości macierzy Q i R podanych w przykładzie z podrozdziału 4.1.1 została przeprowadzona symulacja dla konfiguracji początkowej $x_0 = (0.3, 0, 0, 0, 0, 0)^T$. Wyniki symulacji zostały przedstawione na rysunku 4.4. Można zauważyć, że dla takiego początkowego wektora stanu regulator LQR radzi sobie z utrzymaniem wahadła w punkcie równowagi chwiejnej.



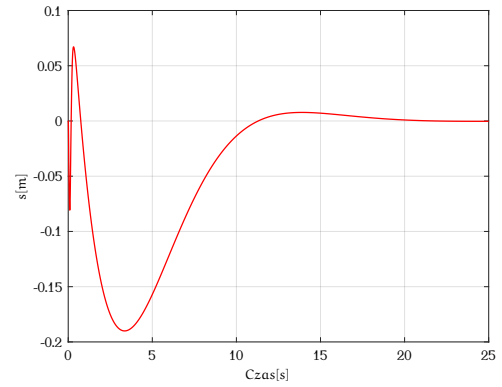
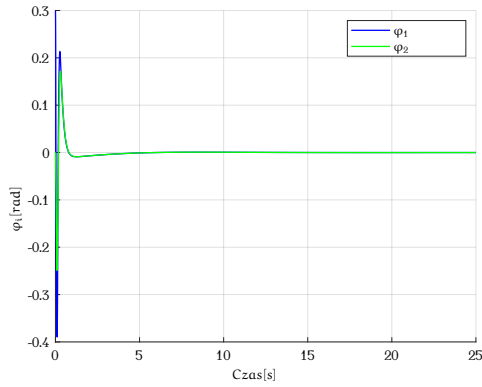
Rysunek 4.4 Przebieg składowych wektora q uzyskanych z zastosowaniem regulatora LQR dla konfiguracji początkowej $x_0 = (0.3, 0, 0, 0, 0, 0)^T$ przy nominalnych wartościach parametrów wahadła



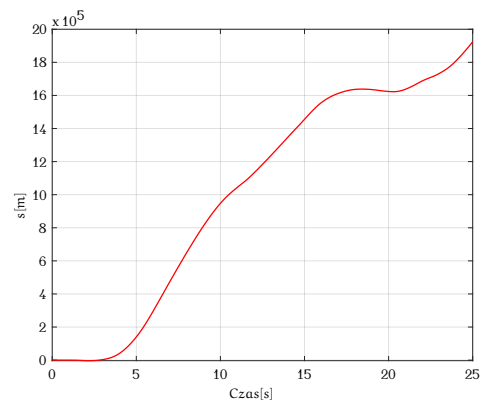
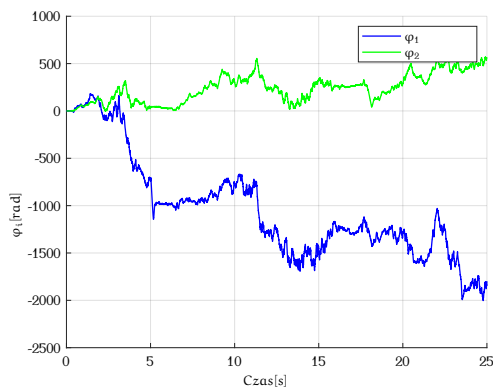
Rysunek 4.5 Przebieg składowych wektora q uzyskanych z zastosowaniem regulatora LQR dla konfiguracji początkowej $x_0 = (0.315, 0, 0, 0, 0, 0)^T$ przy nominalnych wartościach parametrów wahadła

Jeżeli jednak, przy identycznych wartościach parametrów wahadła oraz macierzy Q i R , zostanie zmieniona nieco konfigurację początkową $x_0 = (0.315, 0, 0, 0, 0, 0)^T$ to, jak wynika z wykresów zawartych na rysunku 4.5, regulator LQR przestanie spełniać swoje zadanie.

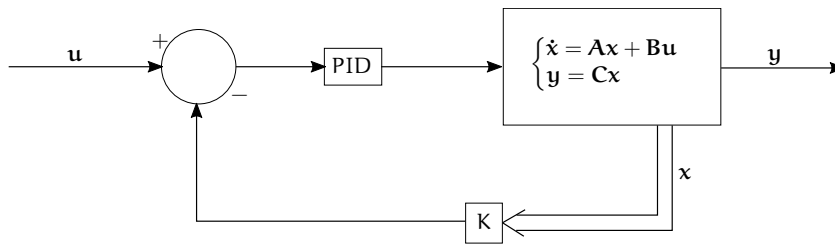
Ostatnie badanie, jakie zostało przeprowadzone, miało na celu pokazanie wpływu parametrów wahadła na początkową konfigurację, przy której regulator LQR potrafi utrzymać wahadło w punkcie równowagi chwiejnej. Rozpatrzmy dwa przypadki, pierwszy z wahadłem o parametrach podanych w tabeli 4.1 i macierzach Q danej równaniem (4.4) oraz $R = 1000$. Drugi z wahadłem o parametrach podanych w tabeli 4.2 oraz identycznych macierzach Q i R jak w pierwszym przypadku. Dla obu przypadków została przeprowadzona symulacja, w której konfiguracja początkowa wynosiła $x_0 = (0.25, 0, 0, 0, 0, 0)^T$. Wyniki symulacji zostały przedstawione na rysunkach 4.6 i 4.7, odpowiednio dla pierwszego oraz drugiego przypadku. Analizując wykresy można zauważyć, że w pierwszym przypadku regulator LQR utrzymuje wahadło w punkcie równowagi chwiejnej, natomiast w drugim nie jest w stanie



Rysunek 4.6 Przebieg składowych wektora \mathbf{q} uzyskanych z zastosowaniem regulatora LQR dla konfiguracji początkowej $\mathbf{x}_0 = (0.25, 0, 0, 0, 0, 0)^T$ przy nominalnym wartościach parametrów wahadła



Rysunek 4.7 Przebieg składowych wektora \mathbf{q} uzyskanych z zastosowaniem regulatora LQR dla konfiguracji początkowej $\mathbf{x}_0 = (0.25, 0, 0, 0, 0, 0)^T$ przy alternatywnych wartościach parametrów wahadła



Rysunek 4.8 Schemat połączenia regulatora LQR oraz PID

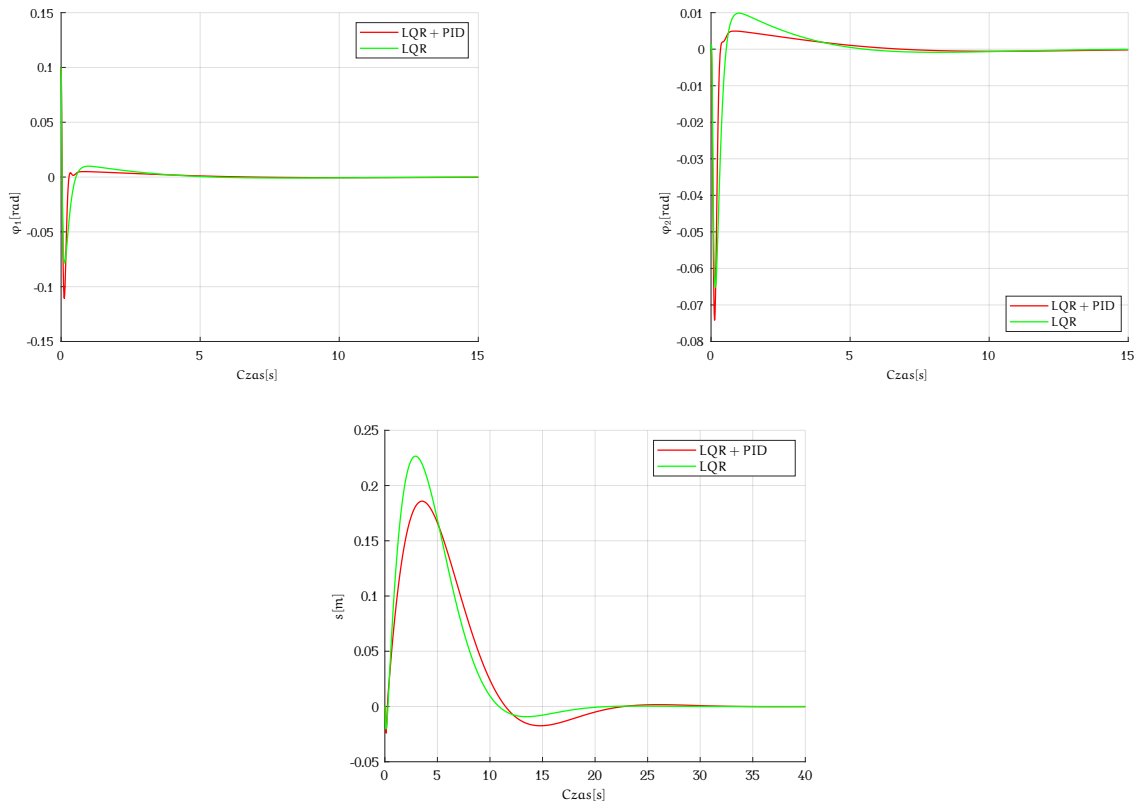
wysterować układ.

4.1.3 Podsumowanie

Na podstawie przeprowadzonych testów można wyciągnąć kilka wniosków dotyczących użycia regulatora LQR jako regulatora dla podwójnego, odwróconego wahadła. Jedną z niedogodności użycia regulatora jest konieczność dobierania wartości macierzy Q oraz R dla konkretnego przypadku metodą prób i błędów. Fakt, że dla jednego wahadła dane wartości pozwalają nim sterować nie oznacza, że pozwolą sterować innym. Kolejnym słabym punktem jest wrażliwość regulatora na większe odchylenia od punktu pracy, dla którego zostało policzone przybliżenie liniowe. W zależności od wartości macierzy Q i R oraz parametrów wahadła, dopuszczalne odchylenia od punktu pracy mogą być różne, jednak regulator nie działa w pełnym zakresie ruchu wahadła, co powoduje że zbyt duże odchylenie będzie prowadzić do niepoprawnej pracy regulatora LQR. Zaletą natomiast jest prosty sposób implementacji regulatora LQR. Macierz sprzężenia K dla danego punktu pracy wystarczy wyliczyć raz przed rozpoczęciem sterowania i nie wymaga on ponownego przeliczania podczas pracy regulatora.

4.2 Regulator LQR z regulatorem PID

Kolejnym sposobem sterowania do punktu podwójnym, odwróconym wahadłem, jaki zostanie przedstawiony, jest połączenie regulatora PID oraz regulatora LQR. Schemat tak uzyskanego układu regulacji został zaprezentowany na rysunku 4.8. W celu implementacji regulatora można postąpić w sposób podobny do podanego w podrozdziale 4.1. Krok jaki należy dodać polega na podaniu sterowania u wyliczonego za pomocą reguły (4.2) na wejście dodatkowego regulatora PID. Wartości otrzymane na wyjściach regulatora PID są nowymi sterowaniami obiektu. Taki układ regulacji w dalszej części pracy będziemy nazywać rozszerzonym regulatorem LQR. Do określenia wartości nastaw PID można użyć metody prób i błędów (taka metoda będzie wykorzystywana w dalszej części pracy). Dobór nastaw będzie wymagał wykonania kilku symulacji pokazujących zachowanie układu regulacji dla różnych wartości nastaw. Na podstawie symulacji będziemy wybierać nastawy, dla których przeregulowania są najmniejsze oraz czas ustawienia wahadła w punkcie równowagi chwiejnej jest najkrótszy.



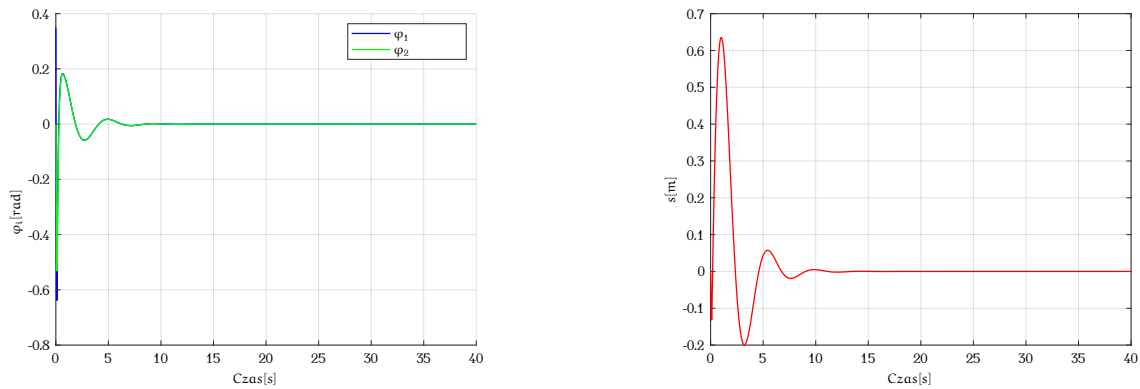
Rysunek 4.9 Porównanie przebiegów składowych wektora q uzyskanych z zastosowaniem regulatora LQR oraz rozszerzonego regulatora LQR przy nominalnych wartościach parametrów wahadła

4.2.1 Porównanie własności regulatora LQR oraz rozszerzonego regulatora LQR

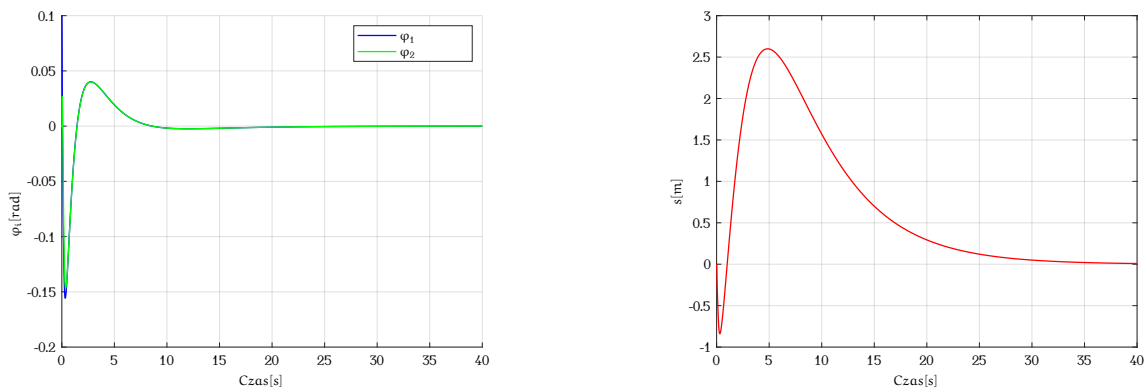
Przedstawiony algorytm rozszerzonego regulatora LQR został zaimplementowany dla podwójnego, odwróconego wahadła o identycznych parametrach jak w przykładzie z podrozdziału 4.1.1. Następnie wykorzystując takie same wartości macierzy Q i R jak w wymienionym przykładzie oraz nastawy PID równe $K_p = 0.7$, $K_i = 0$ i $K_d = 0.0015$, przeprowadzono symulacje z wykorzystaniem pakietu Simulink. Wyniki symulacji zostały przedstawione na rysunku 4.9, gdzie dla porównania pokazano także wyniki symulacji z podrozdziału 4.1.1. Można zauważyć, że dodanie regulatora PID powoduje zmniejszenie zakresu przeregulowań podczas ustawiania wahadła w punkcie równowagi chwiejnej.

4.2.2 Badanie właściwości rozszerzonego regulatora LQR

Pierwszy test będzie miał na celu sprawdzenie czy rozszerzony regulator LQR potrafiysterować układ ustawiony w konfiguracji początkowej, dla której regulator LQR nie był w stanie tego zrobić. Do symulacji wykorzystamy identyczny model wahadła oraz wartości macierzy Q i R jak w poprzednim podrozdziale. W podrozdziale 4.1.2 pokazaliśmy, że regulatorem LQR nie poradził sobie z stabilizacją układu dla



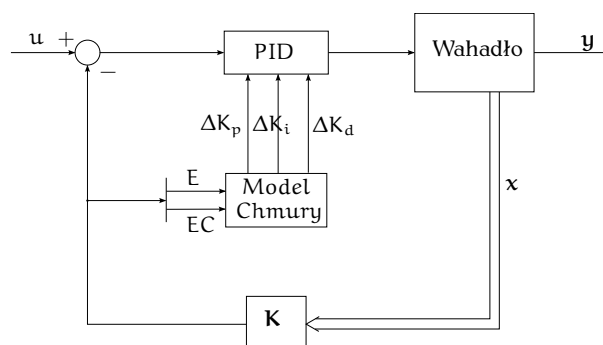
Rysunek 4.10 Przebieg składowych wektora q uzyskanych z zastosowaniem rozszerzonego regulatora LQR dla konfiguracji początkowej $x_0 = (0.35, 0, 0, 0, 0, 0)^T$ przy nominalnych wartościach parametrów wahadła



Rysunek 4.11 Przebieg składowych wektora q uzyskany z zastosowaniem rozszerzonego regulatora LQR przy alternatywnych wartościach parametrów wahadła

początkowej konfiguracji $x_0 = (0.315, 0, 0, 0, 0, 0)^T$. Podczas badania własności rozszerzonego regulatora LQR początkowa konfiguracja wynosiła $x_0 = (0.35, 0, 0, 0, 0, 0)^T$. Nastawy PID zostały ustawione na wartości $K_p = 4$, $K_i = 0$ i $K_d = 0.0015$. Wyniki symulacji zostały przedstawione na rysunku 4.10. Można zauważyć, że rozszerzony regulator LQR był w stanie wysterować układu, mimo większego odchylenia od punktu równowagi chwiejnej. Jednak odchylenie nadal nie może być zbyt duże, żeby regulator działał poprawnie.

Drugi test pozwoli sprawdzić czy rozszerzony regulator LQR potrafi wysterować układ, którego regulator LQR nie był w stanie utrzymać w punkcie równowagi chwiejnej. W symulacji wykorzystano alternatywne wartości parametrów modelu wahadła (tabela 4.1) oraz wartości macierzy Q i R z podrozdziału 4.1.1. Nastawy PID zostały ustawione na wartości $K_p = 2$, $K_i = 0$ oraz $K_d = 0.0015$. Wyniki symulacji przedstawiono na rysunku 4.11. Jak wynika z wykresów, rozszerzony regulator LQR utrzymuje wahadło w stanie równowagi chwiejnej, pomimo że dla takich samych parametrów regulator LQR nie działał poprawnie.



Rysunek 4.12 Schemat regulatora LQR z regulatorem PID o korygowanych nastawach za pomocą modelu chmury (na podstawie [LZG16])

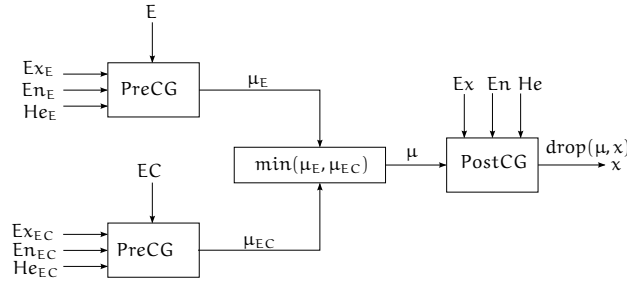
4.2.3 Podsumowanie

Analizując wyniki testów dotyczących rozszerzonego regulatora LQR, można zauważyć, że dodanie regulatora PID pozwala osiągnąć nieco większe początkowe odchylenie wahadła od punktu równowagi chwiejnej, przy których regulator jest w stanie dalej sterować układem. Pozwala również nieco zmniejszyć przeregulowania, jakie pojawiają się podczas początkowej fazy stabilizowania wahadła w punkcie równowagi chwiejnej. Zaletą również jest fakt, że dodanie regulatora PID nie powoduje większej komplikacji w implementacji regulatora. Jako wadę można uznać potrzebę dobrania kolejnych parametrów jakimi są nastawy K_p , K_i i K_d , jednak z drugiej strony ich dobór pozwala nam na większą kontrolę nad sterowaniem układu i lepsze dopasowanie do naszych potrzeb.

4.3 Zastosowanie modelu chmury do zmiany nastaw PID

W podrozdziale pokażemy sposób modyfikacji nastaw regulatora PID z wykorzystaniem modelu chmury. Dokładniejszy opis różnych aspektów związanych z modelem chmury przedstawiono w podrozdziale 2.4. Schemat układu regulacji pokazano na rysunku 4.12. Zgodnie z [LZG16], zmiana nastaw regulatora PID zostaje dokonana na podstawie wartości błędu zintegrowanego E oraz tempa zmian błędu zintegrowanego EC . Po sprawdzeniu do których chmur przynależą aktualne wartości błędów, na podstawie tabel z regułami dotyczącymi dobierania wartości korekcji nastaw K_p , K_i i K_d oraz chmur błędu, sprawdza się do której chmury będzie przynależać wartość korekty nastaw. Mając określone wszystkie te informacje, wartości błędów podajemy na wejścia silnika wnioskowania, którego schemat pokazano na rysunku 4.13. Silnik, na podstawie wartości błędów oraz parametrów chmur, wylicza konkretną wartość korygującą nastawy regulatora PID. Tak skonstruowany układ regulacji w dalszej części będziemy nazywać rozszerzonym regulatorem LQR wykorzystującym model chmury.

Rozszerzony regulator LQR wykorzystujący model chmury można zaimplementować postępując w podobny sposób do zaprezentowanego w podrozdziale 4.2. Zmiana nastaw regulatora PID przy wykorzystaniu modelu chmur wymaga okre-



Rysunek 4.13 Schemat silnika wnioskowania (na podstawie [LZG16])

ślenia tabel reguł, które definiują sposób postępowania przy różnych wartościach błędów wyrażonych w postaci pojęć jakościowych. Takie reguły mogą na przykład mówić, że przy dużych dodatnich błędach E i EC korekcja nastaw K_p powinna być zerowa. Do kompletu, dodatkowo wymagane są chmury zamieniające pojęcia jakościowe związane z błędami i korektami nastaw na konkretne wartości liczbowe.

Pierwszy krok, jaki należy wykonać w realizowanym procesie sterowania, polega na policzeniu wartości błędu zintegrowanego (ang. *integrated error*)

$$E = \sum_{i=1}^n K_i * x_i, \quad (4.5)$$

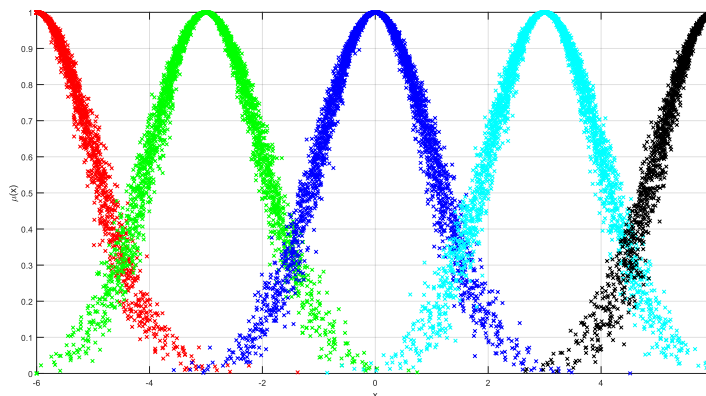
gdzie x_i jest kolejną współrzędną wektora stanu opisującą położenie a K_i podmacierzą macierzy sprzężenia zwrotnego \mathbf{K} , dla odpowiednich współrzędnych x_i , zgodnie z regułą sterowania podaną w równaniu (4.2). Analogicznie należy policzyć tempo zmian błędu zintegrowanego (ang. *integrated error change rate*)

$$EC = \sum_{i=n+1}^{2n} K_i * x_i, \quad (4.6)$$

gdzie x_i jest kolejną współrzędną wektora stanu opisującego prędkość, zaś K_i jest częścią macierzy sprzężenia zwrotnego \mathbf{K} wybraną dla odpowiednich współrzędnych x_i zgodnie z regułą sterowania podaną w równaniu (4.2).

Kolejnym krokiem jest sprawdzenie do których chmur przynależą wyliczone wartości błędów E i EC . Można tego dokonać poprzez wprowadzanie na wejście generatora *precondition cloud generator* (opisanego w części 2.4.3) wartości błędu oraz wartości (Ex, En, He) opisujących chmurę, a następnie sprawdzenie, dla której chmury generator zwrócił największą wartość pewności $\mu(x)$. Wykorzystując chmury do których przynależą błędy oraz tabel z regułami należy wyznaczyć chmury pozwalające wyliczyć wartości liczbowe korekcji nastaw.

Ostatnim krokiem jest wprowadzenie wszystkich informacji do silnika wnioskowania pokazanego na schemacie 4.13. Silnik składa się z dwóch generatorów *PreCG*, do których wprowadzane są błędy E i EC oraz parametry (Ex, En, He) chmur, do których przynależą błędy. Następnie wybierana jest mniejsza wartość z otrzymanych w ten sposób pewności μ_E oraz μ_{EC} i przekazywana na wejścia generatora *postcondition cloud generator* wraz z parametrami chmury odczytanymi z tabeli reguł. W ten sposób zostaje uzyskana wartość korekty nastaw. Korektę należy dodać do wartości nastaw regulatora PID ustalonych na początku działania układu i powrócić do kroku liczenia błędów E i EC dla następnego kroku sterowania.



Rysunek 4.14 Chmury używane do zamiany pojęć jakościowych błędów położenia oraz prędkości na wartości liczbowe

4.3.1 Przykład zastosowania rozszerzonego regulatora LQR wykorzystującego model chmury

Rozszerzony regulator LQR wykorzystujący model chmury został zaimplementowany wykorzystując sposób podany w podrozdziale 4.1.1. Po wyliczeniu wektora wzmocnień sprzężenia zwrotnego \mathbf{K} , został wyliczony błąd zintegrowany

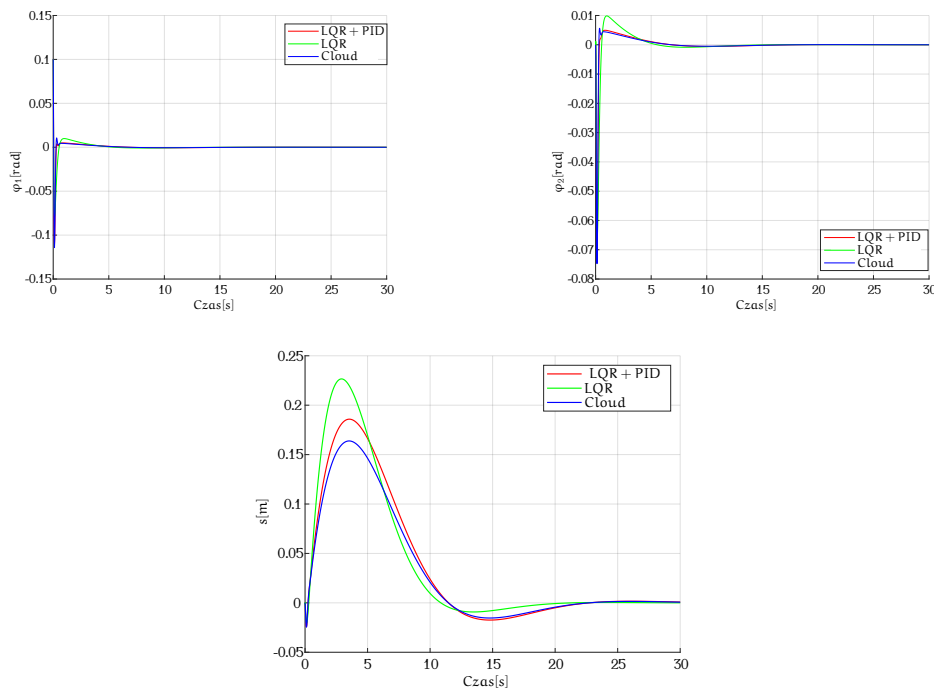
$$E = K_1\varphi_1 + K_2\varphi_2 + K_3s \quad (4.7)$$

oraz tempo zmian błędu zintegrowanego

$$EC = K_4\dot{\varphi}_1 + K_5\dot{\varphi}_2 + K_6\dot{s}. \quad (4.8)$$

Następnie została sprawdzona przynależność wartości błędów do jednej z pięciu chmur: dwóch połowicznych NB (-6, 1, 0.1) i PB (6, 1, 0.1) oraz trzech symetrycznych NS (-3, 1, 0.1), ZE (0, 1, 0.1) i PS (3, 1, 0.1), pokazanych na rysunku 4.14. Na podstawie tabel z regułami 4.3, 4.4 i 4.5 zostają ustalone chmury, do których przynależą poszczególne korekty nastaw. Ponownie wybierana jest jedna z pięciu chmur, takich samych jak przy sprawdzaniu przynależności błędów. Wszystkie informacje dla poszczególnych nastaw zostają podane na wejścia silnika wnioskowania 4.13. Na wyjściu zostają uzyskane wartości korekcji nastaw ΔK_p , ΔK_i oraz ΔK_d , które dodatkowo są przemnażane przez współczynniki 0.01 dla ΔK_p , 0 dla ΔK_i i 0.0001 dla ΔK_d . Współczynniki zostały dobrane na podstawie kilku symulacji, w których wykorzystano różne ich wartości. Występowanie współczynników ma zapobiec pojawieniu się ujemnych wartości nastaw oraz pozwolić na dopasowanie ich korekcji do konkretnego modelu wahadła. Podobny efekt można uzyskać modyfikując parametry (E_x , E_n , H_e) chmur dla poszczególnych korekcji. Ostateczna wartość korygująca nastawy zostaje dodana do początkowych nastaw K_p , K_i i K_d . Dla tak zmodyfikowanych nastaw zostaje wyliczone sterowanie i przekazane na obiekt. W kolejnych krokach sterowania zostają ponownie wyliczone błędy E i EC oraz modyfikacja nastaw.

Rozszerzony regulator LQR wykorzystujący model chmury został zaimplementowany dla modelu wahadła o nominalnych wartościach parametrów. Wartości macierzy \mathbf{Q} i \mathbf{R} oraz początkowych nastaw PID zostały ustawione identycznie jak w pod-



Rysunek 4.15 Porównanie przebiegów składowych wektora q uzyskanych z zastosowaniem regulatora LQR, rozszerzonego regulatora LQR oraz rozszerzonego regulatora LQR wykorzystującego model chmury przy nominalnych wartościach parametrów modelu wahadła

Tabela 4.3 Tabela reguł dla K_p , [LZG16]

E	EC	NB	NS	ZE	PS	PB
NB		NB	NS	NS	NS	ZE
NS		NS	ZE	ZE	PS	PS
ZE		NS	ZE	ZE	PS	PB
PS		NS	ZE	ZE	PS	PS
PB		NB	NS	NS	ZE	ZE

rozdziale 4.2.1. Następnie przeprowadziliśmy symulację, której wyniki zostały przedstawione na rysunku 4.15. Dla porównania na rysunku pokazano również wyniki symulacji z podrozdziałów 4.1.1 i 4.2.1.

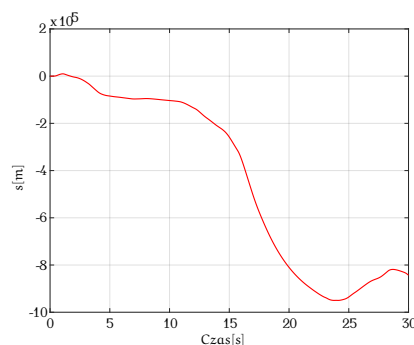
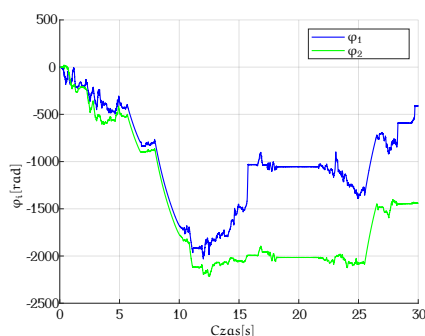
Warto zwrócić uwagę na fakt, że w przedstawionym przykładzie wartość współczynnika, przez który skalowana jest zmiana nastawy dla członu całkującego ΔK_i wynosi 0, co oczywiście powoduje, że nie jest korygowana wartość K_i . Pomimo że w [LZG16] wykorzystano regulator z członem całkującym, po przeprowadzeniu kilku testów, z kilkoma różnymi wartościami współczynnika skalującego, można było dojść do wniosku, że dla tego modelu wahadła, wartości K_i większe od zera nie przyczynia się do zmniejszenia przeregulowań. Powodowały natomiast, że wartość przesunięcia wózka nie zbiegała do zera, a ustalała się na wartości bliskiej zera.

Tabela 4.4 Tabela reguł dla K_i , [LZG16]

E \ EC	NB	NS	ZE	PS	PB
NB	ZE	ZE	ZE	ZE	ZE
NS	PS	PS	PS	ZE	PS
ZE	NS	NS	ZE	PS	NS
PS	NB	ZE	PS	PS	PS
PB	ZE	ZE	ZE	ZE	ZE

Tabela 4.5 Tabela reguł dla K_d , [LZG16]

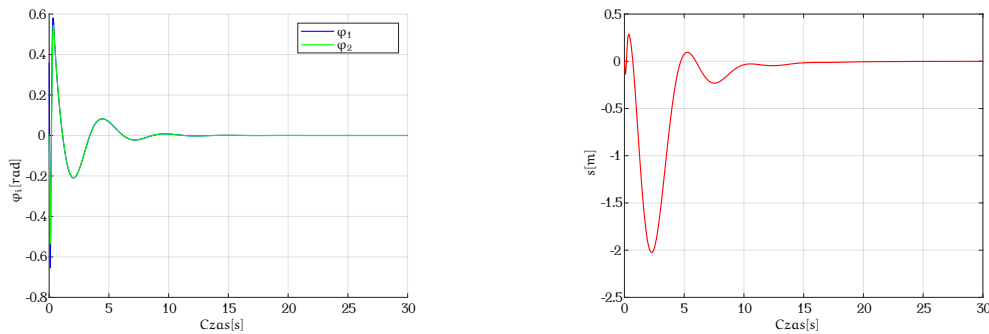
E \ EC	NB	NS	ZE	PS	PB
NB	NS	NB	NB	NB	NS
NS	ZE	NS	NB	NS	ZE
ZE	ZE	NS	NS	NS	ZE
PS	ZE	ZE	ZE	ZE	ZE
PB	PB	PS	PS	PS	PB



Rysunek 4.16 Przebieg składowych wektora q uzyskany z zastosowaniem rozszerzonego regulatora LQR dla konfiguracji początkowej $x_0 = (0.36, 0, 0, 0, 0, 0)^T$ przy nominalnych wartościach parametrów wahadła

4.3.2 Badanie własności rozszerzonego regulatora LQR wykorzystującego model chmury

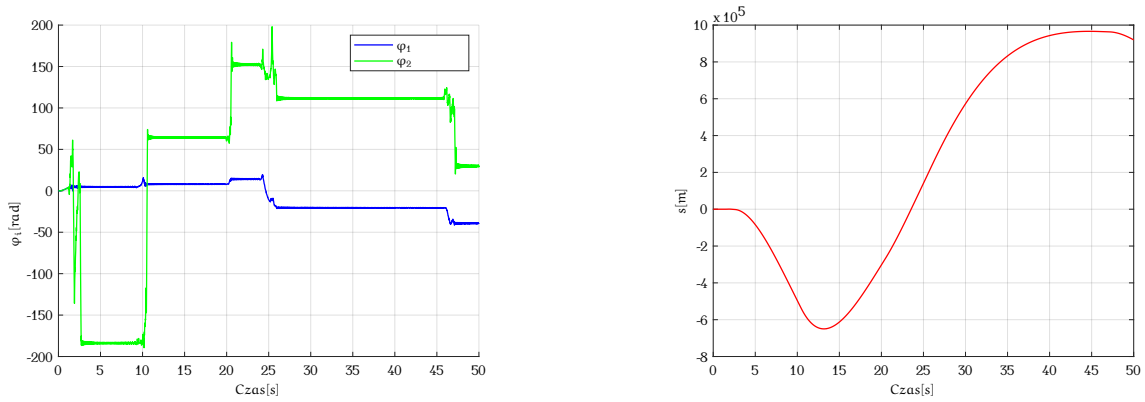
Pierwszy przeprowadzony test miał na celu sprawdzenie, czy rozszerzony regulator LQR wykorzystujący model chmuryysteruje obiekt dla konfiguracji początkowej, dla której rozszerzony regulator LQR nie był w stanie tego zrobić. Dla modelu wahadła przyjęto nominalne wartości parametrów. Przy identycznych wartościach macierzy Q i R oraz początkowych nastaw PID jak w podrozdziale 4.2.1, przeprowadzono dwie symulacje dla konfiguracji początkowej $x_0 = (0.36, 0, 0, 0, 0, 0)^T$. Wyniki pierwszej symulacji, w której wykorzystano rozszerzony regulator LQR, zostały pokazane na rysunku 4.16. Wynika z niej, że regulator nie poradził sobie z wysterowaniem obiektu przy zadanej konfiguracji początkowej. Wyniki drugiej symulacji, uzyskanej przy zastosowaniu rozszerzonego regulatora LQR wykorzystującego mo-



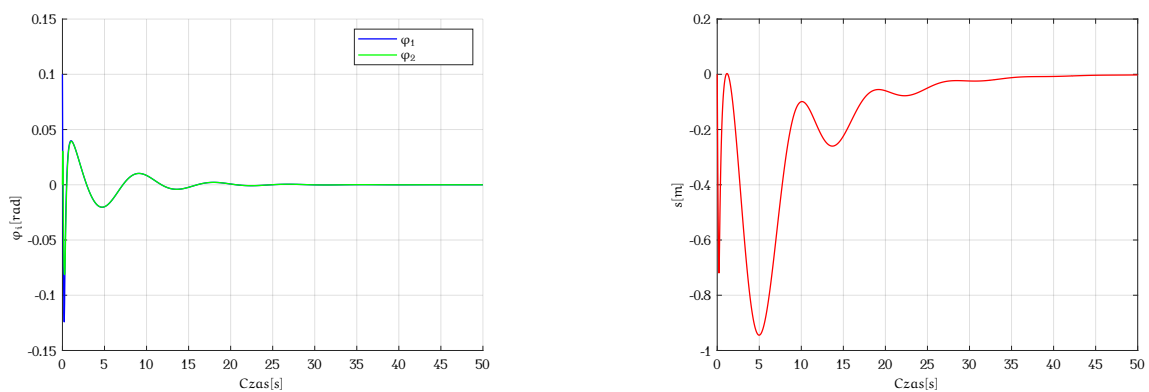
Rysunek 4.17 Przebieg składowych wektora q uzyskany z zastosowaniem rozszerzonego regulatora LQR wykorzystującego model chmury dla konfiguracji początkowej $x_0 = (0.36, 0, 0, 0, 0, 0)^T$ przy nominalnych wartościach parametrów wahadła

del chmury, zostały pokazane na rysunku 4.17. Analizując wyniki drugiej symulacji można stwierdzić, że rozszerzony regulator LQR wykorzystujący model chmury poradził sobie z zadaniem wysterowania obiektu przy założonej konfiguracji początkowej. By tak się stało, współczynniki skalujące ΔK_p ustawiono na 0.6, ΔK_i na 0.3 i ΔK_d na 0.001, a chmury dla nastaw K_i oraz K_d zmieniono na chmury o parametrach $(0, 1, 0.1)$, $(3, 1, 0.1)$, $(6, 1, 0.1)$, $(9, 1, 0.1)$, $(12, 1, 0.1)$. Współczynniki skalujące oraz parametry chmur zostały dobrane na podstawie przebiegu kilku symulacji z różnymi ich wartościami.

Drugi test służył wykazaniu, że rozszerzony regulator LQR wykorzystujący model chmury może wysterować układ, którego regulator LQR oraz rozszerzony regulator LQR nie jest w stanie utrzymać w punkcie równowagi chwiejnej. Do symulacji użyto alternatywne wartości parametrów modelu wahadła podane w tabeli 4.2 oraz wartości macierzy Q i R takie jak w podrozdziale 4.1.1. Konfiguracja początkowa była równa $x_0 = (0.1, 0, 0, 0, 0, 0)^T$. W pierwszym teście, opisanym w podrozdziale 4.1.2, pokazaliśmy, że w takim przypadku regulator LQR nie spełnia swojego zadania. Natomiast test drugi zaprezentowany w podrozdziale 4.2.2 pokazał, że dla nastaw $K_p = 2$, $K_i = 0$ oraz $K_d = 0.0015$ rozszerzony regulator LQR radzi sobie z wysterowaniem wahadła. W tym badaniu ustawimy wartości nastaw $K_p = 0.9$, $K_i = 0$ i $K_d = 0.0015$, dla których rozszerzony regulator LQR nie jest w stanie wysterować układu. Wyniki symulacji wykonanej z zastosowaniem rozszerzonego regulatora LQR zostały przedstawione na rysunku 4.18, natomiast z zastosowaniem rozszerzonego regulatora LQR wykorzystującego model chmury na rysunku 4.19. Wartości współczynników skalujących wynosiły dla ΔK_p 1, dla ΔK_i 0 i dla ΔK_d 0.00025. Chmury dla nastawy K_p zostały zmienione na $(0, 1, 0.1)$, $(3, 1, 0.1)$, $(6, 1, 0.1)$, $(9, 1, 0.1)$, $(12, 1, 0.1)$. Analizując wyniki symulacji możemy stwierdzić, że rozszerzony regulator LQR wykorzystujący model chmury ustabilizował odwrócone, podwójne wahadło w punkcie równowagi chwiejnej, zaś wersji bez chmury nie udało się tego zrobić. Na podstawie testu można wywnioskować, że w pewnych sytuacjach, wykorzystanie modelu chmury do korekcji nastaw regulatora PID pozwala na wysterowanie układu, pomimo że rozszerzony regulator LQR ze źle dobranymi nastawami nie jest w stanie tego zrobić.



Rysunek 4.18 Przebieg składowych wektora \mathbf{q} uzyskany z zastosowaniem rozszerzonego regulatora LQR dla konfiguracji początkowej $\mathbf{x}_0 = (0.1, 0, 0, 0, 0, 0)^T$ przy alternatywnych wartościach parametrów wahadła



Rysunek 4.19 Przebieg składowych wektora \mathbf{q} uzyskany z zastosowaniem rozszerzonego regulatora LQR wykorzystującym model chmury dla konfiguracji początkowej $\mathbf{x}_0 = (0.1, 0, 0, 0, 0, 0)^T$ przy alternatywnych wartościach parametrów wahadła

4.3.3 Podsumowanie

Na podstawie przedstawionych testów można stwierdzić, że rozszerzony regulator LQR wykorzystujący model chmury pozwala na ustabilizowanie podwójnego, odwróconego wahadła w punkcie równowagi chwiejnej dla nieco większego odchylenia początkowego niż rozszerzony regulator LQR. Zastosowanie modelu chmury do korekcji nastaw regulatora PID pozwala także na wysterowanie układu, dla którego rozszerzony regulator LQR z identycznymi nastawami początkowymi nie poradził sobie. Należy jednak zauważyć, że rozszerzony regulator LQR wykorzystujący chmurę ma też swoje ograniczenia i nie zawsze pozwala na tyle poprawić nastawy, aby regulator był w stanie wysterować układ. Ponadto z przeprowadzonych testów wynika, że w przypadku, gdy rozszerzony regulator LQR wykorzystujący model chmury radzi sobie z ustawieniem wahadła w punkcie równowagi chwiejnej, a rozszerzony regulator LQR nie, wystarczy zmienić początkowe nastawy dla rozszerzonego regulatora LQR, aby zaczął spełniać swoje zadanie. Zaletą rozszerzonego regulatora LQR wykorzystującego model chmury jest możliwość zmniejszenia przeregulowań podczas ustawiania wahadła z początkowej konfiguracji do punktu równowagi chwiejnej. Regulator pozwala na dopasowania przebiegów składowych wektora q do własnych potrzeb. Wadą jest jednak konieczność posiadania tabel z regułami zmian nastaw, które można uzyskać jedynie empirycznie. Należy również odpowiednio dobrać wartości parametrów chmur (lub współczynników skalujących) i ich liczbę dla konkretnego obiektu, którym mamy sterować. Wszystko to powoduje, że regulator wykorzystujący model chmury nie jest uniwersalnym narzędziem i wymaga sporej ilości informacji dotyczących obiektu, które często można zebrać tylko poprzez wykonanie serii doświadczeń.

4.4 Podsumowanie

W rozdziale przedstawiliśmy trzy regulatory pozwalające na realizację zadania sterowania do punktu dla podwójnego, odwróconego wahadła. Pierwszym zaprezentowanym regulatorem jest regulator LQR. Jego główną zaletą jest stosunkowo prosta implementacja. Reguła sterowania dla regulatora nie wymaga wielu skomplikowanych obliczeń. Natomiast jako jego wady można uznać konieczność dobierania wartości macierzy Q i R oraz fakt, że regulator LQR potrafi wysterować układ tylko przy małych odchyleniach od punktu pracy, w którym zostało wyliczone przybliżenie liniowe.

Drugim z regulatorów jest rozszerzony regulator LQR. Dodanie regulatora PID do regulatora LQR pozwoliło na zmniejszenie przeregulowań podczas stabilizacji wahadła w punkcie równowagi chwiejnej. Ponadto, dodanie regulatora PID nie wprowadziło większych komplikacji w implementacji systemu regulacji, pozwoliło natomiast na zwiększenie obszaru poprawnej pracy regulatora. Jednakże regulator ciągle nie dla wszystkich możliwych konfiguracji początkowych wahadła. Za wadę można uważać konieczność dobierania kolejnych trzech parametrów, którymi są nastawy K_p , K_i i K_d . Nastawy te należy dopasować do konkretnego obiektu i wymaga to zazwyczaj wykonania kilku prób.

Ostatnim przedstawionym regulatorem jest rozszerzony regulator LQR wyko-

rzystujący model chmury. Jego użycie pozwoliło nieznacznie zwiększyć dopuszczalne początkowe odchylenie, dla którego zachodził poprawny proces regulacji oraz zmniejszyć przeregulowania w porównaniu do rozszerzonego regulatora LQR. Wykorzystanie modelu chmur do modyfikacji nastaw spowodowało jednak duży wzrost komplikacji implementacji regulatora. Chmury użyte w regulatorze muszą zostać odpowiednio dopasowane do konkretnego modelu wahadła. Dodatkowo pojawiły się nowe elementy takie jak tabele z regułami zmiany nastaw, które możemy uzyskać jedynie eksperymentalnie. Powoduje to konieczność zbadania zachowania obiektu przed dobraniem dla niego regulatora LQR wykorzystującego model chmury.

Wszystkie zaprezentowane regulatory wymagają do poprawnego działania dobrania odpowiednich parametrów. Co więcej, parametry te należy najczęściej dobierać metodą prób i błędów. Wymaga to wykonania serii symulacji z różnymi wartościami parametrów oraz porównania ich wyników, w celu wyboru najlepszych dla analizowanego przypadku. Powoduje to dodatkowe trudności podczas próby skorzystania z któregoś z pokazanych regulatorów.

Na podstawie przeprowadzonych testów można dojść do wniosku, że rozszerzony regulator LQR wydaje się najlepszy pod względem stosunku skomplikowania implementacji i czasu, jaki należy poświęcić na dobierania wartości jego parametrów, do jego możliwości objawiających się zakresem wartości początkowego odchylenia wahadła, dla których regulator działa poprawnie oraz wielkościami występujących przeregulowań. Dodanie zmiany nastaw z wykorzystaniem modelu chmury daje niewielki zysk, a powoduje dużą komplikację implementacji. Rozszerzony regulator LQR wykorzystujący model chmury może być jednak przydatny, jeżeli potrzebna będzie większa kontrola nad przebiegami składowych wektora q i zmniejszenie przeregulowań.

Rozdział 5

Generowanie i śledzenie trajektorii

W rozdziale zostanie przedstawiony sposób generowania oraz realizacji trajektorii ruchu dla podwójnego, odwróconego wahadła. Trajektoria powinna rozpoczynać się oraz kończyć w zdefiniowanej konfiguracji wahadła. Na przykład, dla początkowego ustawienia wahadła w punkcie równowagi stabilnej (pionowo w dół, $\varphi_1 = \pi$, $\varphi_2 = \pi$) oraz końcowego w punkcie równowagi chwiejnej (pionowo w górę, $\varphi_1 = 0$, $\varphi_2 = 0$) powinny zostać wygenerowane, dla kolejnych kwantów czasu, wartości wektora stanu x^* wahadła, opisujące ruch między początkową a końcową pozycją. Zadanie to będzie zrealizowane przez opisany poniżej algorytm generowania trajektorii ruchu. Następnie zostanie przedstawiony regulator feedforward wyliczający sterowanie u^* , pozwalające na realizującą wygenerowanej trajektorii. W ostatniej części zostanie pokazany sposób śledzenia wygenerowanej trajektorii. Do tego celu, oprócz regulatora feedforward, zostanie wykorzystany, zaprezentowany w podrozdziale 2.3, regulator LQR.

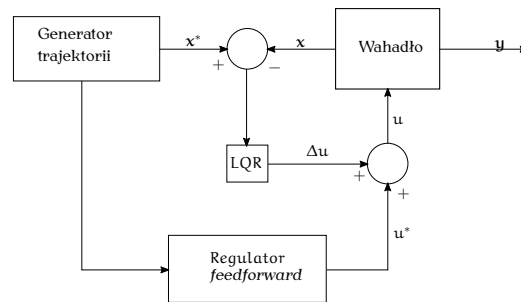
5.1 Generowanie trajektorii ruchu

Wykorzystując opisaną w podrozdziale 2.5 metodę można zaproponować algorytm pozwalający wyznaczyć trajektorie ruchu podwójnego, odwróconego wahadła. Pierwszym krokiem w algorytmie jest dobranie parametrów ograniczeń s_0^\pm , s_1^\pm i s_2^\pm oraz czasu manewru T i parametrów h dla funkcji (2.22).

Kolejny krok polega na znalezieniu rozwiązania układu równań (2.32) przy ograniczeniach (2.28), (2.30) i (2.31) oraz warunkach brzegowych zdefiniowanych przez początkową i końcową konfigurację wahadła, oraz warunki $\xi_{1,0} = \psi_1^{-1}(s_0, \psi_1^\pm)$, $\xi_{1,T} = \psi_1^{-1}(s_T, \psi_1^\pm)$, $\xi_{2,0} = 0$, $\xi_{2,T} = 0$. Należy pamiętać, że nie zawsze przy danych ograniczeniach oraz warunkach brzegowych możliwe jest znalezienie rozwiązania. W takiej sytuacji można próbować zmieniać parametry ustalone w pierwszym kroku algorytmu.

Ostatnim krokiem jest wyliczenie wartości s i \dot{s} na podstawie wzorów (2.23) i (2.24). Otrzymamy wówczas pełny wektor stanu wahadła x^* dla wygenerowanej trajektorii ruchu.

W celu obliczenia sterowania u^* , realizującego wcześniej wygenerowaną trajektorię można zastosować regulator feedforward. Regulator ten do obliczenia sterowania u^* wykorzystuje równanie (2.33) oraz wartości p , ξ_1 i ξ_2 uzyskanych podczas



Rysunek 5.1 Schemat sterowania podwójnego, odwróconego wahadła (na podstawie [GAK13])

generowania trajektorii.

5.2 Śledzenie trajektorii z wykorzystaniem rozszerzonego regulatora feedforward

W podrozdziale zostanie przedstawiony algorytm śledzenia trajektorii ruchu dla odwróconego, podwójnego wahadła. W algorytmie zostanie wykorzystany generator trajektorii, regulator *feedforward* oraz zaprezentowany w podrozdziale 2.3 regulator LQR. Schemat systemu sterowania wahadła został pokazany na rysunku 5.1. W dalszej części pracy taki układ regulacji będziemy nazywać rozszerzonym regulatorem feedforward.

5.2.1 Algorytm śledzenia trajektorii

Obecnie przedstawimy sposób, w jaki wykorzystamy rozszerzony regulator feedforward do śledzenia trajektorii. Pierwszym krokiem jest wygenerowanie trajektorii ruchu dla odwróconego, podwójnego wahadła wykorzystując do tego algorytm podany w podrozdziale 5.1. W efekcie uzyskamy przebieg trajektorii x^* . Kolejno wyliczamy za pomocą regulatora feedforward sterowanie u^* odpowiadające wygenerowanej trajektorii.

Następnym krokiem jest wyliczenie dla kolejnych wartości wektora stanu x^* oraz sterowania u^* wartości macierzy sprzężenia zwrotnego \mathbf{K} , w sposób podobny do przedstawionego w podrozdziale 4.1. Wszystkie powyższe kroki mogą zostać przeprowadzone w trybie *offline*. Ostatnim krokiem jest wyliczenie sterowania zgodnie z równaniami

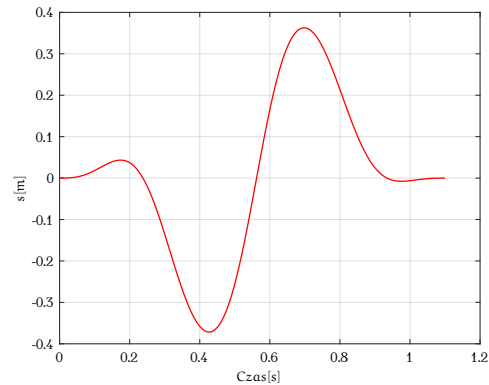
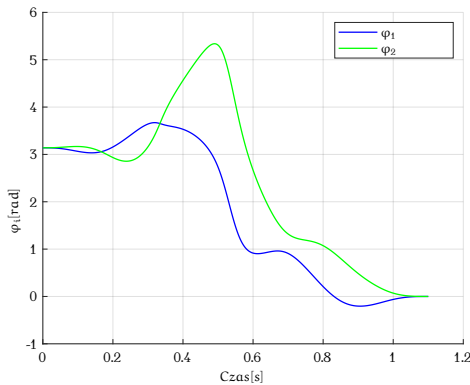
$$\begin{cases} \Delta u = \mathbf{K}(x^* - x) \\ u = u^* + \Delta u \end{cases} \quad (5.1)$$

Ten krok należy już wykonywać w trybie *online*.

5.2. Śledzenie trajektorii z wykorzystaniem rozszerzonego regulatora feedforward⁴¹

Tabela 5.1 Parametry nominalne modelu podwójnego, odwróconego wahadła

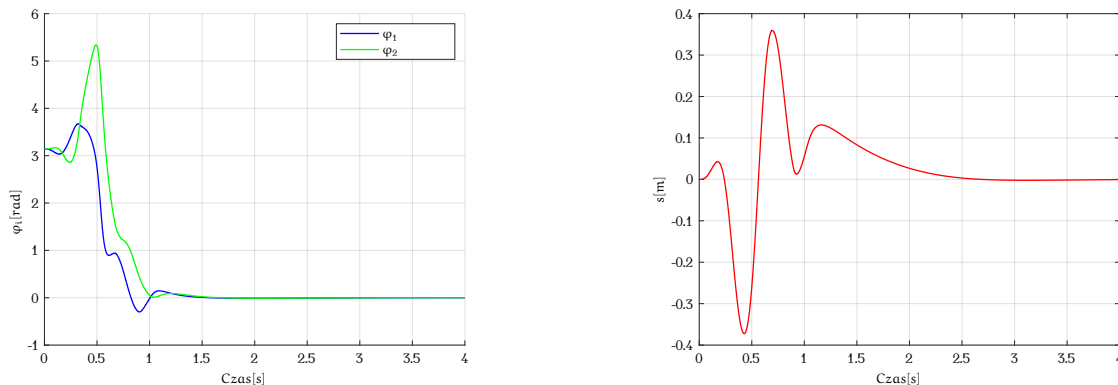
Parametr	Wartość
m_c	1 [kg]
m_1	0.876 [kg]
m_2	0.938 [kg]
l_1	0.323 [m]
l_2	0.419 [m]
d_1	0.215 [Nms]
d_2	0.002 [Nms]



Rysunek 5.2 Wygenerowany przebieg składowych wektora q z zastosowaniem generatora trajektorii ruchu

5.2.2 Zastosowanie rozszerzonego regulatora feedforward do śledzenia trajektorii

Podany w podrozdziale 5.2.1 algorytm śledzenia trajektorii został zaimplementowany dla opisanego w podrozdziale 3.3 modelu wahadła uwzględniającego tarcie. Parametry wahadła zostały przedstawione w tabeli 5.1. Do implementacji oraz symulacji zostało wykorzystane oprogramowanie Matlab z pakietem Simulink. Jako konfigurację początkową przyjęto punkt równowagi stabilnej ($\varphi_1 = \pi$, $\varphi_2 = \pi$ i $s = 0$), natomiast za konfigurację końcową, punkt równowagi chwiejnej ($\varphi_1 = 0$, $\varphi_2 = 0$ i $s = 0$). Pierwszym krokiem było wygenerowanie trajektorii ruchu wahadła między punktami. Zgodnie z algorytmem zostało znalezione rozwiązanie układu równań różniczkowych przy warunkach brzegowych: $s(0) = 0$, $\boldsymbol{\varphi}(0) = (\pi, \pi)^T$, $\dot{s}(0) = 0$, $\dot{\boldsymbol{\varphi}}(0) = \mathbf{0}$, $s(T) = 0$, $\boldsymbol{\varphi}(T) = (0, 0)^T$, $\dot{s}(T) = 0$, $\dot{\boldsymbol{\varphi}}(T) = \mathbf{0}$, $\xi_{1,0} = 0$, $\xi_{1,T} = 0$, $\xi_{2,0} = 0$, $\xi_{2,T} = 0$. Jako ograniczenia przyjęto wartości $s_0^\pm = \pm 0.8$, $s_1^\pm = \pm 10$ i $s_2^\pm = \pm 50$. Do wyliczenia rozwiązania równań użyto funkcji `bvp5c` dostępnej w programie Matlab. Wykresy wartości przesunięcia wózka s oraz kątów φ_1 i φ_2 wygenerowanej trajektorii zostały przedstawione na rysunku 5.2. Czas manewru wynosił $T = 1.1$ oraz parametry $h = (2, 3, 4, 5, 6, 7)$. Na podstawie rozwiązania układu równań różniczkowych zostało policzone sterowanie u^* oraz za pomocą regulatora LQR wartość



Rysunek 5.3 Przebieg składowych wektora q z zastosowaniem rozszerzonego regulatora feedforward

macierzy sprzężenia zwrotnego K dla kolejnych kroków czasu. Wartości macierzy

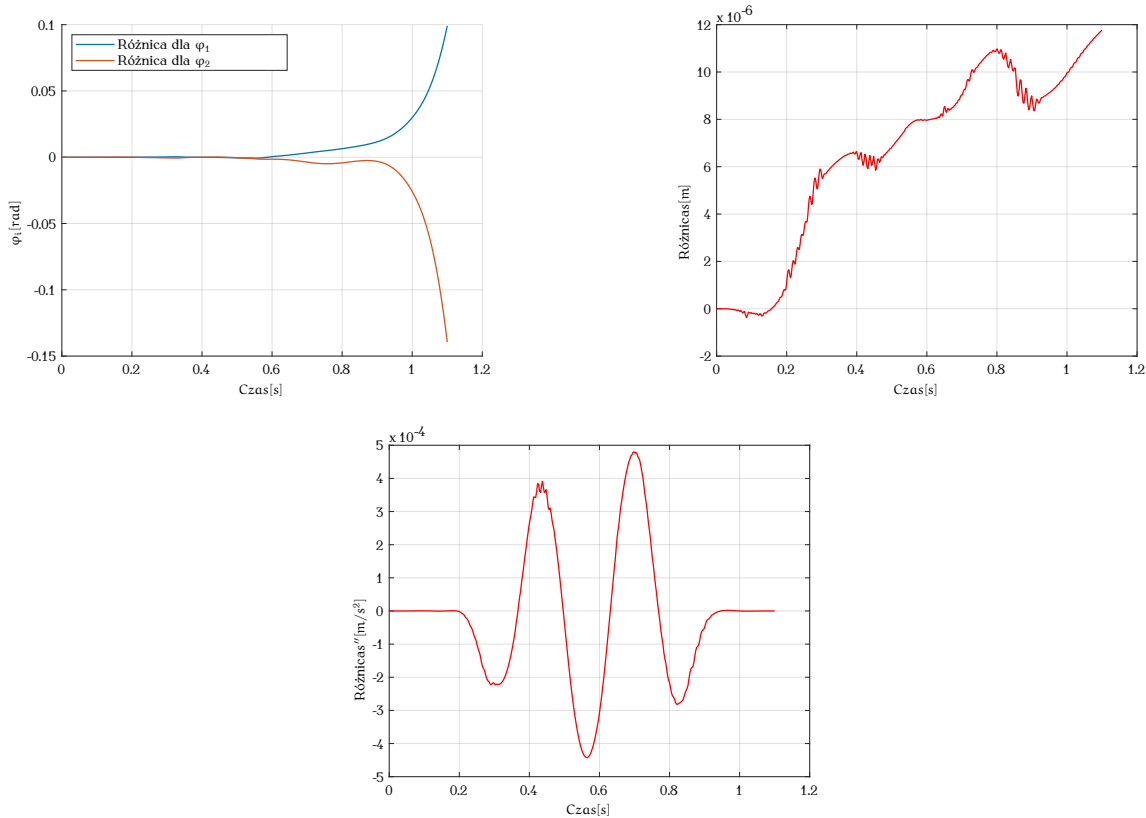
$$Q = \begin{bmatrix} 800 & 0 & 0 & 0 & 0 & 0 \\ 0 & 800 & 0 & 0 & 0 & 0 \\ 0 & 0 & 800 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \quad (5.2)$$

oraz $R = 5$. Wszystkie dane zostały zapisane w tabeli, a następnie wczytane do pakietu Simulink, w którym został zasymulowany układ sterowania pokazany na rysunku 5.1. Wyniki symulacji zostały przedstawione na wykresach 5.3.

Podczas próby zastosowania rozszerzonego regulatora feedforward pojawiło się kilka problemów, o których warto wspomnieć. Pierwszym z nich było zwracanie przez funkcję rozwiązania obciążonego dużym błędem. Dokładność rozwiązania została ustawiona na wartość 10^{-4} jednak raport zwracany po wykonaniu funkcji `bvp5c` wskazywał, że wartość błędu wynosiła w zależności od wartości ograniczeń oraz czasu manewru T i wartości parametrów h od kilku do nawet kilku tysięcy. Aby temu zapobiec, zrezygnowano z warunku brzegowego $\xi_{2,T} = 0$. Można było to zrobić z uwagi na fakt, że w systemie regulacji znajduje się regulator LQR, który pozwala na ustabilizowanie wahadła w punkcie równowagi mimo małych różnic między zadaną końcową konfiguracją, a faktyczną konfiguracją wahadła w czasie T . Należy jednak pamiętać, aby wartości macierzy K dla czasu T były wyliczone dla zadanej konfiguracji końcowej.

Drugi problem pojawił się podczas zastosowania wygenerowanej trajektorii oraz wyliczonych na jej podstawie kolejnych wartości sterowania u^* do sterowania obiektem. Po rozwiązaniu pierwszego problemu, raport generowany przez funkcję `bvp5c` wskazywał na błąd rzędu 10^{-6} . Mimo to, wykonując symulacje można było zauważyć, że trajektorie wahadła przy sterowaniu rozszerzonym regulatorem feedforward były nieco inne niż te wygenerowane. W niektórych przypadkach różnice były zbyt duże, aby regulator LQR poradził sobie z utrzymaniem wahadła w kolejnych punktach trajektorii. Jednak udało się metodą prób i błędów dobrać tak wartości ograniczeń oraz czas manewru T , aby zmniejszyć różnicę między trajekto-

5.2. Śledzenie trajektorii z wykorzystaniem rozszerzonego regulatora feedforward⁴³



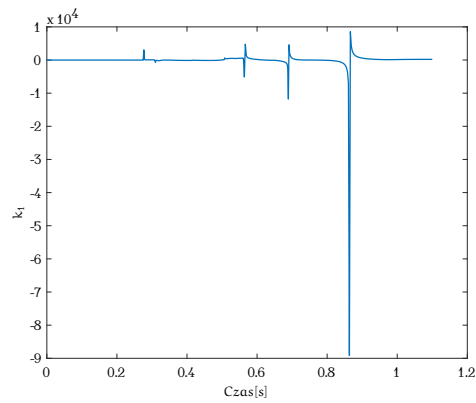
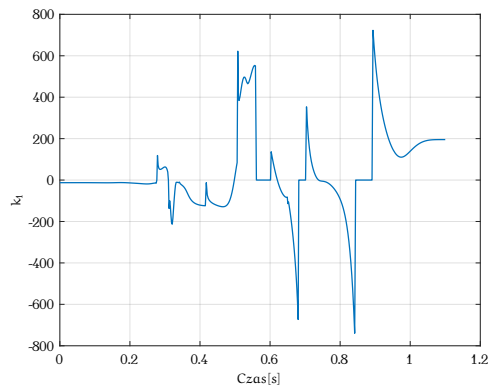
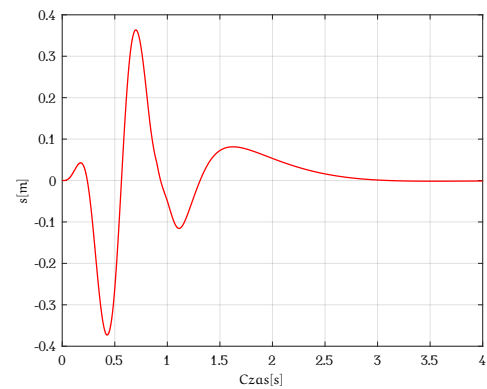
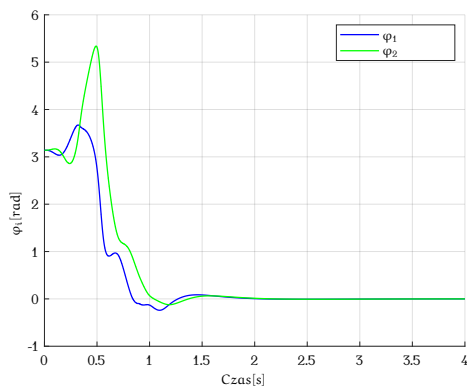
Rysunek 5.4 Różnica między wygenerowaną trajektorią ruchu, a trajektorią ruchu modelu wahadła uzyskaną przy identycznym sterowaniu u^*

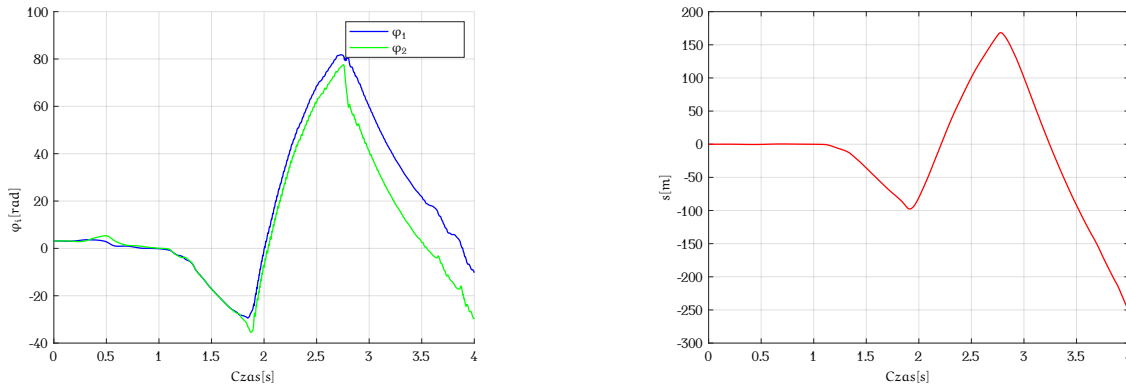
rią wygenerowaną a rzeczywistą do stopnia pozwalającego na niwelację tych różnic za pomocą regulatora LQR. Wykresy różnic między wartościami wygenerowanymi, a wartościami rzeczywistymi składowych wektora q zostały pokazane na rysunku 5.4.

Trzecim problemem były pojawiające się w macierzy K pojedyncze, duże wartości w różnych krokach sterowania. Wykres kolejnych wartości pierwszej składowej k_1 został zaprezentowany na rysunku 5.5. Jak można zauważyć występują tam nagłe skoki wartości, tak jak na przykład w sekundzie 0.86. Tak duże wartości uniemożliwiały w większości przypadków poprawne sterowanie obiektem. Sposobem rozwiązania problemu, z uwagi na pojedyncze występowanie dużych wartości, było ręczne zmniejszenie tych wartości. Wykres tych samych wartości k_1 po modyfikacji został przedstawiony na rysunku 5.6.

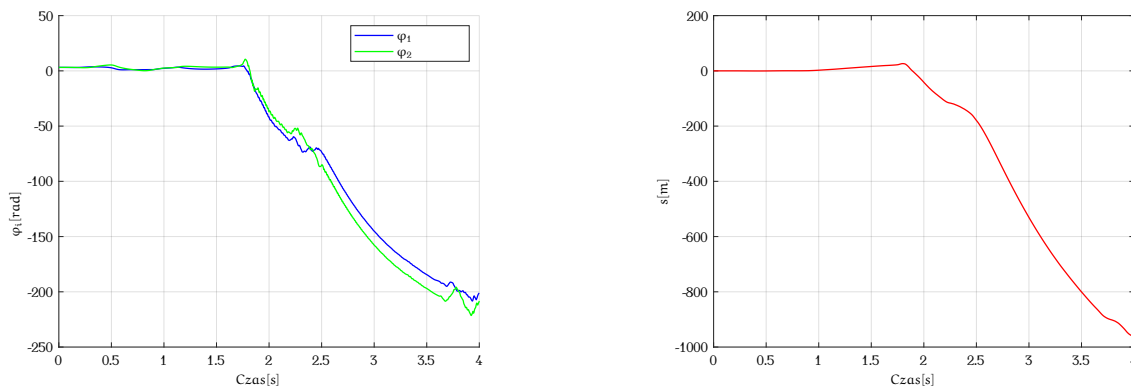
5.2.3 Badanie własności rozszerzonego regulatora feedforward

Testy będą miały na celu sprawdzenie jaki wpływ na rozszerzony regulator feedforward mają niewielkie zaburzenia modelu wahadła. Pierwszy test przeprowadzimy przy zwiększonej masie $m_2 = 0.942[\text{kg}]$. Wykresy przebiegów składowych wektora q zostały pokazane na rysunku 5.7. Analizując wykresy, możemy zauważyć, że regulator przy małym zaburzeniu zrealizował zadaną trajektorię i ustabilizował wahadło w punkcie równowagi chwiejnej. Jednakże dla nieco większej wartości

Rysunek 5.5 Przebieg pierwszej składowej macierzy K Rysunek 5.6 Przebieg pierwszej składowej macierzy K po modyfikacjiRysunek 5.7 Przebieg składowych wektora q z zastosowaniem rozszerzonego regulatora feedforward dla masy $m_2 = 0.942[\text{kg}]$



Rysunek 5.8 Przebieg składowych wektora q z zastosowaniem rozszerzonego regulatora feedforward dla masy $m_2 = 0.943[\text{kg}]$



Rysunek 5.9 Przebieg składowych wektora q z zastosowaniem rozszerzonego regulatora feedforward dla długości drugiego ramienia $l_2 = 0.417[\text{m}]$

$m_2 = 0.943[\text{kg}]$ regulator nie jest w stanie ustabilizować podwójnego, odwróconego wahadła. Zostało to pokazane na rysunku 5.8.

W drugim teście zmienimy natomiast długość drugiego ramienia $l_2 = 0.417[\text{m}]$. Wyniki symulacji zostały zaprezentowane na rysunku 5.9. Z wykresów wynika, że zmiana w modelu długości drugiego ramienia o niewielką wartość powoduje niewłaściwą pracę rozszerzonego regulatora feedforward.

5.3 Podsumowanie

W rozdziale został przedstawiony rozszerzony regulator feedforward, pozwalający na śledzenie trajektorii ruchu przez odwrócone, podwójne wahadło. Uzyskanie poprawnie działającego rozszerzonego regulatora feedforward wymaga odpowiedniego dobrania ograniczeń s_0^\pm , s_1^\pm , s_2^\pm , czasu manewru T oraz wartości parametrów h . Dodatkowo, ponieważ rozszerzony regulator wykorzystuje regulator LQR, należy dobrać odpowiednie wartości macierzy Q i R . Powoduje to konieczność przeprowadzenia serii symulacji sprawdzających poprawność działania rozszerzo-

nego regulatora feedforward dla wygenerowanej trajektorii ruchu oraz dobranych wartościach macierzy. Ponadto, występowanie kilku problemów opisanych w podrozdziale 5.2.2, sprawia, że nie każda wygenerowana trajektoria ruchu, mimo że wydawała się poprawną, pozwalała na ustabilizowanie wahadła w punkcie równowagi chwiejnej.

Przeprowadzone testy pokazują, że rozszerzony regulator feedforward nie jest w stanie sobie poradzić z większymi zaburzeniami modelu. Może powodować to komplikację podczas próby wykorzystania algorytmu do sterowania rzeczywistym obiektem. Z uwagi na dużą czułość na zaburzenia, rozszerzony regulator będzie wymagał dokładnej identyfikacji poszczególnych wartości parametrów modelu oraz niezmiennie warunki otoczenia.

Rozdział 6

Podsumowanie i wnioski

Celem pracy było dokonanie przeglądu algorytmów sterowania podwójnym, odwróconym wahadłem, implementacja wybranych z nich w środowisku symulacyjnym oraz zbadanie ich własności. Wszystkie te zadania zostały zrealizowane w pełni. Spośród podanych we wstępie prac, opisujących różne sposoby sterowania wahadłem, zostały wybrane dwie [LZG16] i [GAK13]. Następnie zostały przytoczone podstawowe zagadnienia matematyczne, opis regulatora LQR, przedstawiono zagadnienia związane z modelem chmury oraz sposób generowania trajektorii ruchu dla wahadła. Kolejno w pracy został wyprowadzony model matematyczny podwójnego, odwróconego wahadła, wykorzystany do symulacji zachowania wahadła podczas implementacji poszczególnych regulatorów.

W rozdziale 4 zostały zaprezentowane sposoby sterowania do punktu wahadła. Ich działanie zilustrowano przedstawiając przykładowe zastosowanie regulatora LQR i rozszerzonego regulatora LQR, a także rozszerzonego regulatora LQR wykorzystującego model chmury przedstawionego w [LZG16] do sterowania wahadłem. Wszystkie te regulatory poddano badaniom właściwości oraz porównano ze sobą. W rozdziale 5 został podany algorytm pozwalający na śledzenie trajektorii ruchu wahadła. Wykorzystując sposób generowania trajektorii oraz system regulacji zaprezentowany w [GAK13] pokazano sposób zastosowania rozszerzonego regulatora feedforward do śledzenia trajektorii przez wahadło. Ponadto przeprowadzono testy mające na celu sprawdzenie możliwości rozszerzonego regulatora feedforward.

Praca miała przede wszystkim przedstawić różne algorytmy sterowania podwójnym, odwróconym wahadłem w sposób pozwalający na ich łatwe odtworzenie przez czytelnika. Przeprowadzone testy poszczególnych sposobów sterowania pozwoliły na wskazanie wad oraz zalet ich użycia. W pracy zostały również opisane różne problemy, które wystąpiły podczas implementacji, czy też w trakcie testów poszczególnych regulatorów oraz metody radzenie sobie z nimi. Pozwala to traktować pracę jako wstęp teoretyczny do budowy sterownika podwójnego, odwróconego wahadła. Praca pozwala czytelnikowi na stwierdzenie, czy przedstawione algorytmy sterowania nadają się do realizacji zadania w jego przypadku.

Przeprowadzone badania prowadzą do wniosku, że przed zastosowaniem przedstawionych algorytmów sterowania do rzeczywistego wahadła konieczne jest wykonywanie symulacji. I tak regulatory LQR potrzebują do właściwego działania odpowiednio dobranych macierzy Q i R . Niewłaściwe wartości tych macierzy mogą powodować duże przeregulowania, a nawet niestabilność systemu regulacji. Ponad-

to często dobór wartości parametrów wymaga zastosowania metody prób i błędów. Wszystko to powoduje, że próba implementacji algorytmu na rzeczywistym obiekcie bez wykonania wcześniejszych symulacji i dobrania parametrów, może być niebezpieczna i spowodować nawet uszkodzenie obiektu.

Literatura

- [GAK13] T. Glück, A. Eder, A. Kugi, Swing-up control of a triple pendulum on a cart with experimental validation. *Automatica*, 49(3):801–808, 2013.
- [KCHI17] S. Khatoon, D. K. Chaturvedi, N. Hasan, M. Istiyaque, Optimal control of a double inverted pendulum by linearization technique. W: *International Conference on Multimedia, Signal Processing and Communication Technologies (IMPACT)*, strony 123–127, 2017.
- [LD05] D.Y. Li, Y. Du, *Artificial intelligence with uncertainty*. Beijing: National Defence Industry Press, 2005.
- [LZG16] H. Li, M. Zhang, C. Guo, Cloud-model pid control of double inverted pendulum based on information fusion. W: *35th Chinese Control Conference (CCC)*, strony 574–578, Chengdu, China, 2016.
- [Mur06] R. M. Murray. Lecture 2 – LQR control. <https://www.cds.caltech.edu/~murray/courses/cds110/wi06/lqr.pdf>, 2006.
- [QDFY11] Q. Qiana, D. Dongmei, L. Feng, T. Yongchuan, Stabilization of the double inverted pendulum based on discrete-time model predictive control. W: *IEEE International Conference on Automation and Logistics (ICAL)*, strony 243–247, 2011.
- [Raj16] S. Raj, Reinforcement learning based controller for stabilization of double inverted pendulum. W: *IEEE 1st International Conference on Power Electronics, Intelligent Control and Energy Systems (ICPEICES)*, strony 1–5, 2016.
- [SK08] B. Siciliano, O. Khatib, *Handbook of Robotics*. Springer, 2008.
- [TM17] K. Tchoń, R. Muszyński, *Metody matematyczne automatyki i robotyki. Notatki do wykładu z dziedziny automatyki i robotyki*. Katedra Cybernetyki i Robotyki, Wydział Elektroniki, Politechnika Wrocławska, 2017.
- [TM18a] K. Tchoń, R. Muszyński, *Mechanika analityczna. Notatki do wykładu z dziedziny automatyki i robotyki*. Katedra Cybernetyki i Robotyki, Wydział Elektroniki, Politechnika Wrocławska, 2018.
- [TM18b] K. Tchoń, R. Muszyński, *Robotyka. Notatki do wykładu z dziedziny automatyki i robotyki*. Katedra Cybernetyki i Robotyki, Wydział Elektroniki, Politechnika Wrocławska, 2018.
- [TMD⁺00] K. Tchoń, A. Mazur, I. Dulęba, R. Hossa, R. Muszyński, *Manipulatory i Roboty Mobilne modele, planowanie ruchu, sterowanie*. Akademicka Oficyna Wydawnicza PLJ, 2000.

- [TNY03] M. Takahashi, T. Narukawa, K. Yoshida, Intelligent control using destabilized and stabilized controllers for a swung up and inverted double pendulum. W: *IEEE International Symposium on Intelligent Control*, strony 914–919, 2003.

Spis tabel

4.1	Parametry nominalne modelu podwójnego, odwróconego wahadła . .	22
4.2	Alternatywne wartości parametrów modelu podwójnego, odwróconego wahadła	23
4.3	Tabela reguł dla K_p , [LZG16]	33
4.4	Tabela reguł dla K_i , [LZG16]	34
4.5	Tabela reguł dla K_d , [LZG16]	34
5.1	Parametry nominalne modelu podwójnego, odwróconego wahadła . .	41

Spis rysunków

2.1	Schemat regulatora LQR	7
2.2	Przykładowa chmura wygenerowana dla wartości $E_x = 0$, $E_n = 1$ oraz $H_e = 0.1$ (na podstawie [LZG16])	9
2.3	Różne rodzaje chmur: a) symetryczna, b), c) połowiczna, d) asymetryczna [LD05]	10
2.4	Schematy generatorów chmury: a) forward cloud generator, b) backward cloud generator [LD05]	11
2.5	Schematy generatorów chmury: a) precondition cloud generator, b) postcondition cloud generator (na podstawie [LD05])	12
2.6	Generator zasady z pojedynczym warunkiem (na podstawie [LD05])	13
2.7	Generator zasady z podwójnym warunkiem (na podstawie [LZG16])	13
3.1	Schemat odwróconego, podwójnego wahadła	18
4.1	Przykładowy przebieg wektora konfiguracji $\mathbf{q} = (\varphi_1, \varphi_2, s)^T$ uzyskany z zastosowaniem regulatora LQR	23
4.2	Przebieg składowych wektora \mathbf{q} przy alternatywnych wartościach parametrów wahadła i niezmiennych wartościach regulatora LQR	24
4.3	Przebieg składowych wektora \mathbf{q} przy alternatywnych wartościach parametrów wahadła i zmienionych wartościach regulatora LQR	24
4.4	Przebieg składowych wektora \mathbf{q} uzyskanych z zastosowaniem regulatora LQR dla konfiguracji początkowej $\mathbf{x}_0 = (0.3, 0, 0, 0, 0, 0)^T$ przy nominalnych wartościach parametrów wahadła	25
4.5	Przebieg składowych wektora \mathbf{q} uzyskanych z zastosowaniem regulatora LQR dla konfiguracji początkowej $\mathbf{x}_0 = (0.315, 0, 0, 0, 0, 0)^T$ przy nominalnych wartościach parametrów wahadła	25
4.6	Przebieg składowych wektora \mathbf{q} uzyskanych z zastosowaniem regulatora LQR dla konfiguracji początkowej $\mathbf{x}_0 = (0.25, 0, 0, 0, 0, 0)^T$ przy nominalnym wartościach parametrów wahadła	26
4.7	Przebieg składowych wektora \mathbf{q} uzyskanych z zastosowaniem regulatora LQR dla konfiguracji początkowej $\mathbf{x}_0 = (0.25, 0, 0, 0, 0, 0)^T$ przy alternatywnych wartościach parametrów wahadła	26
4.8	Schemat podłączenia regulatora LQR oraz PID	27
4.9	Porównanie przebiegów składowych wektora \mathbf{q} uzyskanych z zastosowaniem regulatora LQR oraz rozszerzonego regulatora LQR przy nominalnych wartościach parametrów wahadła	28

4.10	Przebieg składowych wektora q uzyskanych z zastosowaniem rozszerzonego regulatora LQR dla konfiguracji początkowej $x_0 = (0.35, 0, 0, 0, 0, 0)^T$ przy nominalnych wartościach parametrów wahadła	29
4.11	Przebieg składowych wektora q uzyskany z zastosowaniem rozszerzonego regulatora LQR przy alternatywnych wartościach parametrów wahadła	29
4.12	Schemat regulatora LQR z regulatorem PID o korygowanych nastawach za pomocą modelu chmury (na podstawie [LZG16])	30
4.13	Schemat silnika wnioskowania (na podstawie [LZG16])	31
4.14	Chmury używane do zamiany pojęć jakościowych błędów położenia oraz prędkości na wartości liczbowe	32
4.15	Porównanie przebiegów składowych wektora q uzyskanych z zastosowaniem regulatora LQR, rozszerzonego regulatora LQR oraz rozszerzonego regulatora LQR wykorzystującego model chmury przy nominalnych wartościach parametrów modelu wahadła	33
4.16	Przebieg składowych wektora q uzyskany z zastosowaniem rozszerzonego regulatora LQR dla konfiguracji początkowej $x_0 = (0.36, 0, 0, 0, 0, 0)^T$ przy nominalnych wartościach parametrów wahadła	34
4.17	Przebieg składowych wektora q uzyskany z zastosowaniem rozszerzonego regulatora LQR wykorzystującego model chmury dla konfiguracji początkowej $x_0 = (0.36, 0, 0, 0, 0, 0)^T$ przy nominalnych wartościach parametrów wahadła	35
4.18	Przebieg składowych wektora q uzyskany z zastosowaniem rozszerzonego regulatora LQR dla konfiguracji początkowej $x_0 = (0.1, 0, 0, 0, 0, 0)^T$ przy alternatywnych wartościach parametrów wahadła	36
4.19	Przebieg składowych wektora q uzyskany z zastosowaniem rozszerzonego regulatora LQR wykorzystującym model chmury dla konfiguracji początkowej $x_0 = (0.1, 0, 0, 0, 0, 0)^T$ przy alternatywnych wartościach parametrów wahadła	36
5.1	Schemat sterowania podwójnego, odwróconego wahadła (na podstawie [GAK13])	40
5.2	Wygenerowany przebieg składowych wektora q z zastosowaniem generatora trajektorii ruchu	41
5.3	Przebieg składowych wektora q z zastosowaniem rozszerzonego regulatora feedforward	42
5.4	Różnica między wygenerowaną trajektorią ruchu, a trajektorią ruchu modelu wahadła uzyskaną przy identycznym sterowaniu u^*	43
5.5	Przebieg pierwszej składowej macierzy K	44
5.6	Przebieg pierwszej składowej macierzy K po modyfikacji	44
5.7	Przebieg składowych wektora q z zastosowaniem rozszerzonego regulatora feedforward dla masy $m_2 = 0.942[\text{kg}]$	44
5.8	Przebieg składowych wektora q z zastosowaniem rozszerzonego regulatora feedforward dla masy $m_2 = 0.943[\text{kg}]$	45
5.9	Przebieg składowych wektora q z zastosowaniem rozszerzonego regulatora feedforward dla długości drugiego ramienia $l_2 = 0.417[\text{m}]$	45