



POLITECHNIKA BIAŁOSTOCKA

WYDZIAŁ MECHANICZNY

TEMAT

**ANALIZA PORÓWNAWCZA PRZEBIEGÓW FUNKCJI LAPUNOWA
UKŁADÓW DYNAMICZNYCH CAŁKOWITYCH I NIECAŁKOWITYCH
RZĘDÓW**

Wykonawca: Dawid Ostaszewicz

Opiekun: dr inż. Krzysztof Rogowski

BIAŁYSTOK, 2025 ROK

Karta dyplomowa

POLITECHNIKA BIAŁOSTOCKA Wydział Mechaniczny	studia stacjonarne drugiego stopnia	nr albumu 109382
		rok akademicki 2024/2025
		kierunek studiów Automatyka i robotyka
		specjalność Automatyka przemysłowa

inż. Dawid Ostaszewicz
(imiona i nazwisko studenta)

TEMAT PRACY DYPLOMOWEJ:

Analiza porównawcza przebiegu funkcji Lapunowa układów dynamicznych całkowitych i niecałkowitych rzędów

Zakres pracy:

1. Stabilność układów dynamicznych wg Lapunowa
2. Analiza stabilności układów całkowitego rzędu z wykorzystaniem funkcji Lapunowa
3. Analiza stabilności układów niecałkowitego rzędu z wykorzystaniem funkcji Lapunowa
4. Analiza numeryczna przebiegów funkcji Lapunowa dla wybranych przypadków układów dynamicznych
5. Wnioski końcowe

Słowa kluczowe (max 5):

- stabilność
- funkcja Lapunowa
- trajektoria
- układy niecałkowitego rzędu

Krzysztof Rogowski, dr inż.
(imiona i nazwisko, stopień/tytuł opiekuna pracy dyplomowej) - podpis

ABSTRACT

This study focuses on the stability analysis of dynamic systems of integer and fractional orders based on Lyapunov functions. This method is widely used in control theory and dynamic system modeling. The research includes a broad theoretical analysis as well as practical applications of Lyapunov function synthesis methods for systems of different orders.

The first part presents fundamental concepts related to the Lyapunov method and the stability of dynamic systems. Classical dynamical description formalisms, such as Newtonian, Lagrangian, and Hamiltonian approaches, are discussed, along with concepts related to the optimization of energy functions in linear systems. Special attention is given to the optimization of the quadratic Lyapunov function and its application in stability analysis.

Next, the theoretical foundations of fractional-order calculus, which enables the modeling of memory-dependent systems, are described. Definitions of fractional-order calculus, including the Riemann-Liouville, Caputo, and Grünwald-Letnikov approaches, are presented. Additionally, numerical tools such as MATLAB and Simulink are introduced, enabling the implementation of fractional-order systems and their stability analysis using Lyapunov functions.

In the following section, a comparison of Lyapunov functions for integer- and fractional-order systems is presented. Three methods for synthesizing Lyapunov functions for different order systems were developed and tested: the classical Lyapunov function method – applying traditional Lyapunov functions to fractional-order systems, the Simplex optimization method – based on numerical optimization of the cost functional, and the conditional Lyapunov function optimization method – focused on determining Lyapunov functions that satisfy the negative derivative criterion.

The conducted numerical experiments allowed for evaluating the effectiveness of each method in the context of an algorithm for stabilizing switched systems. It was shown that the appropriate selection of a Lyapunov function for fractional-order systems is crucial for effective stability analysis and system control.

STRESZCZENIE

Niniejsza praca dotyczy analizy stabilności układów dynamicznych o całkowitych i niecałkowitych rzędach w oparciu o funkcje Lapunowa. Metoda ta jest powszechnie stosowana w teorii sterowania i modelowaniu systemów dynamicznych. Praca obejmuje szeroką analizę teoretyczną oraz zastosowania praktyczne metod syntezy funkcji Lapunowa dla układów o różnych rzędach.

W pierwszej części przedstawiono podstawowe pojęcia związane z metodą Lapunowa oraz stabilnością układów dynamicznych. Omówiono klasyczne formalizmy opisu dynamiki, takie jak podejście Newtonowskie, Lagrange'a oraz Hamiltona, a także wprowadzono pojęcia związane z optymalizacją funkcji energii układów liniowych. Szczególną uwagę poświęcono optymalizacji kwadratowej funkcji Lapunowa oraz jej zastosowaniu w analizie stabilności układów.

Następnie opisano teoretyczne podstawy rachunku niecałkowitego rzędu, który pozwala na modelowanie układów z pamięcią. Przedstawiono definicje rachunku niecałkowitego rzędu, w tym podejścia Riemanna-Liouville'a, Caputo oraz Grünwalda-Letnikova. Dodatkowo zaprezentowano narzędzia numeryczne, takie jak MATLAB i Simulink, które umożliwiają implementację układów niecałkowitego rzędu oraz analizę ich stabilności w kontekście funkcji Lapunowa.

W kolejnej części pracy zaprezentowano porównanie funkcji Lapunowa dla układów całkowitych i niecałkowitych rzędów. Opracowano oraz przetestowano trzy metody syntezy funkcji Lapunowa dla układów o różnym rzędzie: metoda klasycznej funkcji Lapunowa – zastosowanie tradycyjnych funkcji Lapunowa w układach o niecałkowitym rzędzie, metoda optymalizacji Simpleks – bazująca na numerycznej optymalizacji funkcjonału kosztu oraz metoda warunkowej optymalizacji funkcji Lapunowa – polegająca na wyznaczaniu funkcji Lapunowa spełniającej kryterium ujemnej pochodnej. Przeprowadzone eksperymenty numeryczne pozwoliły na ocenę skuteczności poszczególnych metod w kontekście analizy algorytmu stabilizacji układów przełączalnych. Wykazano, że odpowiedni dobór funkcji Lapunowa dla układów niecałkowitego rzędu jest istotny dla skutecznej analizy stabilności oraz sterowania układami.

SPIS TREŚCI

1 WSTĘP	2
1.1 CEL I ZAKRES PRACY	2
1.2 PRZEGŁĄD LITERATURY	3
1.3 METODYKA REALIZACJI	5
1.4 CHARAKTERYSTYKA PRACY	5
2 STABILNOŚĆ UKŁADÓW DYNAMICZNYCH	7
2.1 WPROWADZENIE	7
2.2 MODELOWANIE UKŁADÓW KLASYCZNYCH	8
2.3 FORMALIZMY OPISU DYNAMIKI	10
2.4 OPTYMALIZACJA LINIOWO-KWADRATOWA	11
2.5 BEZPOŚREDNIA METODA LAPUNOWA	14
2.6 PODSUMOWANIE ROZDZIAŁU	18
3 ANALIZA STABILNOŚCI UKŁADÓW NIECAŁKOWITEGO RZĘDU	19
3.1 WPROWADZENIE	19
3.2 DEFINICJE RACHUNKU NIECAŁKOWITEGO RZĘDU	19
3.3 REPREZENTACJE NUMERYCZNE W MATLAB	21
3.4 NUMERYCZNA I ANALITYCZNA POCHODNA FUNKCJI HEAVISIDE'A	22
3.5 WARUNKI POCZĄTKOWE	26
3.6 TRANSLACJA W PRZESTRZENI STANU	29
3.7 FUNKCJE LAPUNOWA UKŁADÓW UŁAMKOWYCH	30
3.8 OPTYMALNE FUNKCJE LAPUNOWA - METODA SIMPLEKS	34
3.9 OPTYMALIZACJA WARUNKOWA FUNKCJI LAPUNOWA	39
3.10 FUNKCJE LAPUNOWA NA PRZEDZIALE CZASU	41
3.11 PODSUMOWANIE ROZDZIAŁU	48
4 STABILIZACJA UKŁADÓW PRZEŁĄCZALNYCH	49
4.1 WSTĘP DO ROZDZIAŁU	49
4.2 UKŁADY PRZEŁĄCZALNE	49
4.3 PROBLEM STABILIZACJI	51
4.4 OPRACOWANIE ALGORYTMU STABILIZACJI	53
4.5 BADANIE ALGORYTMU STABILIZACJI	54
4.6 PODSUMOWANIE ROZDZIAŁU	58
5 PODSUMOWANIE PRACY	60

1. WSTĘP

Metody badań stabilności oparte na funkcjach Lapunowa stanowią podstawowe i formalne narzędzie analizy stabilności układów dynamicznych. Ich zastosowanie pozwoliło na wyprowadzenie wielu znanych dziś analitycznych metod badania stabilności systemów stacjonarnych, liniowych oraz całkowitego rzędu. Dzięki wysokiej skuteczności metod analitycznych, w przypadku wymienionych układów nie ma już konieczności bezpośredniego stosowania funkcji Lapunowa. Jednak metoda ta znajduje nadal szerokie zastosowanie w algorytmach sterowania optymalnego oraz w analizie układów niestacjonarnych, nieliniowych i niecałkowitego rzędu. Podstawowe badania dotyczące stabilności układów oraz odpowiedzi na fundamentalne pytania w tej dziedzinie stanowią klucz do dalszego rozwoju teorii sterowania. Proces ten przebiega od analizy prostych przypadków aż po uogólnienia obejmujące złożone struktury dynamiczne.

Praca stanowi odpowiedź na pytania badawcze postawione w ramach zespołu realizującego projekt finansowany przez Narodowe Centrum Nauki pt. „*Analiza i synteza układów sterowania z wykorzystaniem układów przełączalnych niecałkowitego rzędu*”, kierowanego przez prof. T. Kaczorkę; nr 2022/45/B/ST7/03076. Autor niniejszej pracy jest członkiem zespołu realizującego wymieniony grant. Badania przedstawione w pracy obejmują szeroki zakres zagadnień – od analizy zbieżności definicji układów niecałkowitego rzędu, poprzez fizyczne aspekty stosowalności warunków początkowych w rachunku niecałkowitych rzędów, aż po opracowanie algorytmu stabilizacji układów przełączalnych niecałkowitego rzędu oraz metod pozyskiwania funkcji Lapunowa układów niecałkowitych rzędów i ich porównanie w ramach opracowanego algorytmu.

1.1. CEL I ZAKRES PRACY

Celem pracy jest analiza porównawcza przebiegów funkcji Lapunowa dla układów całkowitego i niecałkowitego rzędu, prowadząca do opracowania metod doboru funkcji Lapunowa dla układów niecałkowitego rzędu oraz ich porównania w opracowanym w ramach pracy algorytmie stabilizacji układów przełączalnych. Zakres pracy dzieli się na pięć etapów, które obejmują:

- opis podstawowych zagadnień związanych z bezpośrednią metodą Lapunowa,
- analizę procesu badania stabilności metodą funkcji Lapunowa w wypadku układów całkowitych rzędów,
- opracowanie i analizę metod odtwarzania funkcji Lapunowa układów niecałkowitych rzędów oraz analizę ich stabilności,

- analizę numeryczną przebiegów funkcji Lapunowa dla wybranych przypadków układów dynamicznych.

1.2. PRZEGŁAD LITERATURY

Z zakresu badań stabilności układów całkowitego rzędu klasyczną pracą jest "*Stability of Motion*" [18], w której omówiono zagadnienia stabilności układów zarówno metodą bezpośrednią Lapunowa, jak i innymi podejściami. Praca ta przedstawia fundamentalne zagadnienia, powszechnie stosowane w analizie stabilności układów liniowych oraz nieliniowych całkowitego rzędu. Kolejną istotną pozycję literatury dotyczącą badania stabilności układów dynamicznych, a także innych fundamentalnych zagadnień teorii sterowania, jest "*Teoria Sterowania i Systemów*" [1]. Szczególnie ważne w niej jest przedstawienie faktu, że algebraiczne równanie Riccatiego prowadzi do analitycznego wyznaczenia parametrów kwadratowej funkcji Lapunowa. Prawa fizyki, na podstawie których można definiować modele elektrycznych układów dynamicznych są omówione w "*Teorii Obwodów Elektrycznych*" [2], natomiast podstawowe zasady formułowania modeli w oparciu o ogólne prawa fizyki zestawia "*Control Systems Engineering*" [3].

Definicje oraz omówienie rachunku niecałkowitego rzędu zostały przedstawione w pracy "*Fractional Linear Systems and Electrical Circuits*" [28]. Natomiast wybrane zagadnienia rachunku ułamkowego rzędu zestawiono w "*Selected Problems of Fractional Systems Theory*" [19]. Praca ta zawiera szczególnie istotne aspekty dotyczące odtwarzania dyskretnej definicji rachunku, która – wraz z pracą "*The Grünwald-Letnikov formula and its equivalent Horner's form accuracy comparison and evaluation for application to fractional order PID controllers*" [48] – stanowi podstawę numerycznej implementacji rachunku niecałkowitego rzędu.

W literaturze opisywane są szerokie zastosowania rachunku niecałkowitych rzędów. Praca "*Fractional models of falling object with linear and quadratic frictional forces considering Caputo derivative*" [33] przedstawia zastosowanie rachunku o rzędach niecałkowitych do modelowania siły tarcia obiektu w swobodnym spadku. Istnieją jednak poważne zastrzeżenia w zastosowanym modelu, ponieważ siła w myśl definicji przedstawionej przy zmianie pochodnej z rzędu całkowitego na niecałkowity traci swoją jednostkę, przez co nie można dokonać budowy modelu przedstawionego w tej pracy, ponieważ brak jest fizycznych przesłanek, albo przekształcenia, które odbuduje siłę wyrażoną w Newtonach, jako efekt końcowy przekształceń, dzięki czemu można byłoby zastosować wspomniany rachunek w przypadku analizowanego modelu. Z drugiej strony poprawnym przykładem zastosowania omawianego rachunku jest "*A delayed plant disease model with Caputo fractional derivatives*" [32], w którym omawiany jest model rozprzestrzeniania się choroby wśród roślin. Model jest fizycznie poprawny, ponieważ wielkości wektora stanu wyrażają liczby np. zainfekowanych roślin, przez co są bez jednostki i fizycznej interpretacji oraz w tym przypadku odwołanie do cechy rachunku, którą jest pamięć jako odwzorowanie w pewnym sensie układu odpornościowego wydaje się być traf-

nym spostrzeżeniem i zastosowaniem modeli ułamkowych rzędów. Sprawa podobnie kształtuje się w przypadku "*Modeling and analysis of sustainable approach for dynamics of infections in plant virus with fractal fractional operator*" [29]. Podobne problemy z jednostkami i fizycznymi reprezentacjami są powielane w wielu pracach [24, 31], również przy wykorzystywaniu rachunku w ramach układów sterowania np. FOPID (Fractional order proportional-integral-derivative). Przykładem są regulatory stosowane w układach regulacji automatycznej. W standardowym zastosowaniu przetwarzają sygnał uchybu regulacji pod postacią różnicy wielkości fizycznej, w odpowiedzi na wyjściu zwracając sygnał sterujący pod postacią wielkości sterującej układem. Jeśli stosowany jest regulator FOPID przekształcenia jednostek miar w większości układów i zastosowań tracą sens zastosowania, bez odpowiedniego układu, który niejako by dekodował wielkości otrzymywane z regulatora niecałkowitego rzędu na wielkości znane fizycznie i obecne w wektorze stanu modelu układu dynamicznego. Prawie zgodny z zasadami fizyki jest model przedstawiony w pracy "*Modeling and Fuzzy FOPID Controller Tuned by PSO for Pneumatic Positioning System*" [30]. Zastosowanie modelu rozmytego wydaje się być odpowiednim krokiem w procesie swoistego dekodowania wielkości otrzymywanych w regulatorze FOPID na wielkości fizycznie poprawnie zawarte w wektorze stanu. Jednak autorzy zastosowali w pracy odwrotną kolejność, najpierw stosując rozmycie i proces odwrotny odzyskując wielkości fizyczne, później stosując FOPID, który już nie odtwarza poprawnie wielkości fizycznych. Jednak w myśl takiego toku rozumowania klasyczne regulatory PID powinny być równie bezużyteczne jak FOPID, ponieważ w wypadku definicji równoległej sygnał wyjściowy jest sumą różnych jednostek, co formalnie nie powinno mieć miejsca. Powstają jednak prace przeglądowe "*FOPID Controllers and Their Industrial Applications: A Survey of Recent Results I*" [22], które świadczą o tym, że regulatory FOPID są coraz częściej stosowane w przemyśle usprawiedliwiając zwiększone zużycie zasobów obliczeniowych na rzecz zmniejszonych kosztów osiągnięcia sterowania. Należy jednak jednocześnie zwracać uwagę na formalizmy fizyczne i możliwości interpretacji fizycznych aspektów modeli, co przedstawia [21]. W bieżącej pracy modele występują w formie ogólnego zastosowania, bez przypisanych jednostek, w związku z czym wymienione problemy nie mają miejsca. Mogłoby się wydawać, że jedynym problemem byłoby odszukanie obiektów reprezentowanych przez rzędy niecałkowite, jednak fizycznie można je odtworzyć do różnych zastosować przy udziale układów drabinkowych, których odpowiedź jest zgodna z podstawowymi definicjami rachunku [47], obiekty tego rodzaju mogą również występować w informatyce pod postacią cyfrową, modelując ataki DDoS [43, 44, 42, 41].

Funkcje Lapunowa układów o rzędach niecałkowitych są używane w pracy "*Lyapunov functions for fractional order systems*" [34]. Metoda polega poszukiwaniu kandydatów na funkcję wśród funkcji niecałkowitych rzędów, co stanowi fizyczny problem, którego nie można pomijać, ponieważ funkcje Lapunowa skorelowane są z energią układów i Hamiltonianem. W klasycznym układzie Hamiltonian jest rzędu całkowitego. Proponowane są rozwiązania optymalizacji które bazują na równaniach Euler'a-Lagrange'a niecałkowitego rzędu [25], jednak takie

przedstawienie Lagrangianu oraz Hamiltonianu jest kontrowersyjne ze względu na pochodne cząstkowe niecałkowitego rzędu, po wymiarach przestrzeni stanu. Pojawia się problem z interpretacją fizyczną efektu pamięci odnoszącego się do wymiarów przestrzeni jak i czasu. Brak jest merytorycznych i fizycznych przesłanek za zastosowaniem tego rodzaju operacji. Z wymienionego względu funkcja Lapunowa układu niecałkowitego rzędu powinna pozostać w reprezentacji i naturze całkowita, dzięki czemu pomimo stosowania układów niecałkowitych rzędów właściwości fizyczne i korelacja energetyczna pozostaje zgodna z fundamentalnymi zasadami fizyki. Główną problematyką bieżącej pracy jest opracowanie metod doboru funkcji Lapunowa układów niecałkowitych rzędów, zgodnych z podstawowymi zasadami fizyki oraz weryfikacja ich stosownalności.

1.3. METODYKA REALIZACJI

Głównym celem badań jest analiza porównawcza przebiegów funkcji Lapunowa układów całkowitych i niecałkowitych rzędów, która prowadzi do opracowania metod syntezy funkcji Lapunowa układów niecałkowitych rzędów, zgodnej z fundamentalnymi zasadami fizyki oraz formalnymi właściwościami bezpośredniej metody Lapunowa. W pracy pominięto problem odwzorowania wielkości fizycznych w układach niecałkowitych rzędów przez stosowanie modelowania w podejściu ogólnym, bez wykorzystania jednostek fizycznych. Takie podejście zostało zastosowane w związku z ograniczoną liczbą modeli fizycznych ułamkowych rzędów, natomiast modele układów o rzędach niecałkowitych mogą być również wykorzystywane w modelach, których wektor stanu zawiera wielkości ilościowe.

Realizacja zadań związanych z pracą odbywa się w środowisku obliczeniowym języka Matlab [6] oraz Simulink [14]. Modele układów niecałkowitego rzędu są odtwarzane przy użyciu bibliotek Matlaba [11, 4, 27, 15]. Biblioteki te wymagają dodatkowych pakietów narzędziowych [12, 13]. Niektóre przykłady odnoszące się do wizualnej reprezentacji formalnych definicji zostały przygotowane przy użyciu języka Python [9] wraz z bibliotekami obsługującymi wizualizację i przetwarzanie danych [7, 8, 10].

Porównanie otrzymanych metod syntezy funkcji Lapunowa niecałkowitego rzędu odbywa się na opracowanym w ostatnim rozdziale algorytmie stabilizacji układu przełączalnego całkowitych i niecałkowitych rzędów. Kryterium porównawczym w wymienionym wypadku jest całkowy wskaźnik jakości, który pozwala ocenić jakość sygnału otrzymanego na wyjściu układu, podczas osiągania stanu ustalonego.

1.4. CHARAKTERYSTYKA PRACY

Drugi rozdział stanowi teoretyczny wstęp do pracy, przedstawiając podstawową terminologię związaną z metodami wykorzystywanyimi podczas badań i prac nad syntezą funkcji Lapunowa układów niecałkowitych rzędów. Przedstawia podstawowe definicje występujące w teorii sterowania i modelowaniu układów dynamicznych wraz z różnymi formalizmami opisu dy-

namiki. Następnie wprowadzony jest problem optymalizacji związany z zasadą minimalnego działania, która jest nieodzownym elementem związanym z syntezą kwadratowej funkcji Lapunowa liniowych, stacjonarnych układów całkowitego rzędu. Kolejnym elementem rozdziału jest wprowadzenie bezpośredniej metody Lapunowa oraz wyjaśnienie podstawowych definicji z nią związanych.

Rozdział trzeci zawiera przegląd podstawowych definicji rachunku niecałkowitego rzędu, ze szczególnym uwzględnieniem jego zastosowania w modelowaniu układów dynamicznych. Omówiono zarówno definicje ciągłe, jak i dyskretnie, analizując ich wzajemną zbieżność. W dalszej części rozdziału przedstawiono również oryginalny wkład autora — zaproponowano trzy metody syntezy funkcji Lapunowa dla układów niecałkowitego rzędu, stanowiące podstawę późniejszych analiz stabilności.

Rozdział czwarty stanowi zasadniczą część wkładu własnego. Skoncentrowano się w nim na problemie stabilizacji układów przełączalnych zarówno całkowitego, jak i niecałkowitego rzędu. Przeprowadzono badania jakościowe i ilościowe zaproponowanych metod syntezy funkcji Lapunowa, porównując ich skuteczność w różnych klasach układów. Wyniki te stanowią podstawę wniosków końcowych pracy.

2. STABILNOŚĆ UKŁADÓW DYNAMICZNYCH

2.1. WPROWADZENIE

Stabilność układów dynamicznych stanowi jedno z fundamentalnych zagadnień w teorii sterowania i modelowania systemów, a jej analiza jest nieodzowna w zrozumieniu zachowania układów w różnych obszarach nauki i techniki. Zagadnienie to jest odpowiedzią na następujące pytanie, czy układ dąży do stanu równowagi dynamicznej, czy też jego trajektorie mogą w sposób niekontrolowany oddalać się od niej. Analiza stabilności jako istotne narzędzie, dostarcza również wiedzy na temat reakcji systemu w środowiskach zakłócienniowych, zarówno zewnętrznych, jak i wynikających z jego wewnętrznej struktury.

Jednym z podejść do analizy stabilności jest metoda Lapunowa, która opiera się na formalnym badaniu właściwości trajektorii układu. W klasycznym ujęciu teoria Lapunowa pozwala badać, czy dany układ dynamiczny wykazuje stabilność asymptotyczną, stabilność w sensie ogólnym lub niestabilność, na podstawie matematycznych reprezentacji funkcji energii. W niniejszym rozdziale przedstawiono fundamentalne założenia stabilności układów dynamicznych w sensie Lapunowa, rozpoczynając od analizy klasycznych modeli liniowych i stacjonarnych. Modele te są następnie rozwijane i reprezentowane matematycznie w przestrzeni stanu, co pozwala na zastosowanie bardziej zaawansowanych technik analizy [18].

Rozdział przedstawia także analityczną procedurę doboru odpowiednich kandydatów na funkcję Lapunowa. Układy dynamiczne, podobnie jak inne procesy fizyczne, podporządkowują się uniwersalnym zasadom, takim jak zasada minimalnego działania. W naturze trajektorie układów dążą do ekstremalizacji energetycznej, wybierając ścieżki o najniższym możliwym wydatku energii, realizowanym w najkrótszym czasie. To zachowanie, w połączeniu z determinizmem modeli liniowych i stacjonarnych, umożliwia dopasowanie funkcji energii do naturalnego funkcjonału kosztu [1].

Podrozdziały koncentrują się na szczegółowym omówieniu bezpośredniej metody Lapunowa. W pierwszej kolejności przedstawiono podstawowe definicje i zależności, które pozwalają zrozumieć, na czym polega działanie funkcji Lapunowa i jakie są jej główne zastosowania. Następnie opisano metodę ułatwiającą identyfikację potencjalnych kandydatów na funkcję Lapunowa, co jest ważnym etapem analizy stabilności układów dynamicznych. Dodatkowo w tekście przybliżono rys historyczny oraz genezę funkcjonowania tej metody, ukazując jej naturalne piękno i zgodność z zasadą minimalnego działania.

2.2. MODELOWANIE UKŁADÓW KLASYCZNYCH

Układy liniowe, stacjonarne, z jednym wejściem o całkowitym rzędzie są opisywane równaniem:

$$\mathcal{D}x_i(t) = \sum_{j=1}^n a_{ij}x_j(t) + b_i u(t) \quad i = 1, 2, \dots, n \quad , \quad (2.1)$$

gdzie:

- $\mathcal{D} = \frac{d}{dt}$ – różniczka,
- $n \in \mathbb{Z}^+$ – rząd układu,
- $x(t) = [x_1, x_2, \dots, x_n]^T \in X \subseteq \mathbb{R}^n$ – wektor stanu,

$$\bullet A = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nn} \end{bmatrix} \in \mathbb{R}^{n \times n} \text{ – macierz stanu,}$$

- $B = [b_1, b_2, \dots, b_n]^T \in \mathbb{R}^n$ – wektor wejścia,
- $u(t) \in U \subseteq \mathbb{R}$ – sterowanie,
- U – zbiór sterowań dopuszczalnych,

modelujących dynamikę systemów należących do tej klasy. Rozwiążanie równania stanu, spełniające warunek początkowy, zawiera dwie składowe: swobodną i wymuszoną:

$$x(t) = e^{A(t-t_0)}x_0 + \int_{t_0}^t e^{A(t-\tau)}Bu(\tau)d\tau \quad , \quad (2.2)$$

gdzie:

- $t_0 \in T \subseteq \mathbb{R}$ – chwila początkowa,
- $t \in T \subseteq \mathbb{R}^+$ – bieżąca chwila,
- $x_0 \in X \subseteq \mathbb{R}^n$ – wektor warunków początkowych.

Na podstawie podanej postaci rozwiązania równania stanu można stwierdzić, że jeśli wyłącznie składowa swobodna jest stabilna, wtedy ciągły, liniowy, stacjonarny układ, poddany sterowaniu również będzie stabilny. Przedstawione wnioskowanie pozwala na redukcję poziomu skomplikowania badania stabilności opisywanych systemów do postaci autonomicznej:

$$x(t) = e^{A(t-t_0)}x_0. \quad (2.3)$$

Jeśli trajektoria układu autonomicznego rozpoczynająca się z warunku początkowego jest stabilna, oznacza to, że dany układ ciągły, liniowy i stacjonarny jest globalnie stabilny, co znaczy, że pozostaje stabilny niezależnie od warunków początkowych i wymuszeń.

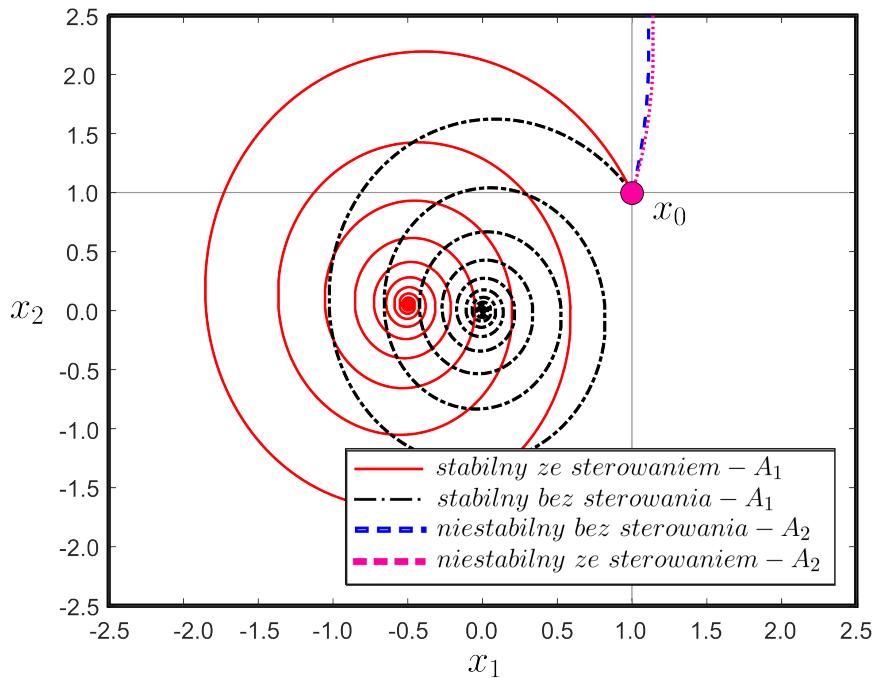
PRZYKŁAD 2.1

Należy rozważyć układy drugiego rzędu (2.3) o podanych macierzach stanu:

$$A_1 = \begin{bmatrix} -0.1 & -1 \\ 2 & -0.1 \end{bmatrix}, \quad (2.4)$$

$$A_2 = \begin{bmatrix} 2 & -1 \\ 2 & 2 \end{bmatrix}, \quad (2.5)$$

wektorze wymuszeń $B = [0 \ 1]^T$; warunkach początkowych $x_0 = [1, \ 1]^T$ i sterowaniu $u(t) = \mathbb{1}(t)$, $u = 0 \cdot \mathbb{1}(t)$ – układ poddany i niepoddany sterowaniu.



Rysunek 2.1: Porównanie trajektorii stabilnego i niestabilnego układu: z wymuszeniem i bez wymuszenia

Rozwiązania przedstawione na Rysunku 2.1 wskazują, że jeśli trajektoria układu autonomicznego zmierza do zera, układ jest stabilny. Niemniej jednak, ta metoda nie jest powszechnie stosowana w praktyce inżynierskiej, ponieważ badania eksperymentalne na drogich systemach nie byłyby opłacalne. Z tego powodu w praktyce częściej stosuje się modelowanie matematyczne oraz różnorodne kryteria służące do analizy stabilności ciągłych, liniowych układów dynamicznych, które szczegółowo opisuje prof. T. Kaczorek [1].

2.3. FORMALIZMY OPISU DYNAMIKI

Mechanika klasyczna w kontekście fizyki ciał stałych, dopuszcza kilka jednoznacznych podejść do analizy układów. Pierwszym z nich jest metoda Newtonowska. Opiera się na prawach dynamiki Newtona, szczególnie na drugiej zasadzie:

$$F = ma = m\mathcal{D}^2x, \quad (2.6)$$

gdzie:

- F – siła wypadkowa,
- m – masa,
- a – przyspieszenie,
- x – położenie,

mówiącej, że jeżeli na ciało działa stała siła wypadkowa, to ciało porusza się ruchem jednostajnie przyspieszonym; z przyspieszeniem wprost proporcjonalnym do działającej siły, a odwrotnie proporcjonalnym do masy ciała. Po sformułowaniu równań różniczkowych opisujących dynamikę układu, czy to na gruncie mechaniki, czy teorii obwodów (I, II prawo Kirchhoffa [2]), możliwe jest przejście do uogólnionej reprezentacji w przestrzeni stanu.

Drugie podejście ma swoje podstawy w optymalizacji oraz fundamentalnej zasadzie Hamiltona. Zakłada ona, że natura zawsze wybiera trajektorię, która ekstremalizuje (zazwyczaj minimalizuje) określoną wielkość fizyczną zwaną działaniem. W przypadku teorii sterowania ta zasada oznacza, że trajektoria układów autonomicznych również jest ekstremalą, która minimalizuje jednostkową (naturalną) funkcję kosztu [5]:

$$\min \mathcal{S} = \int_{t_1}^{t_2} L dt, \quad (2.7)$$

gdzie:

- \mathcal{S} – działanie,
- L – Lagrangian,

przy czym Lagrangian jest funkcją, która opisuje energię układu:

$$L(x(t), \dot{x}(t), t) = \mathcal{T}(x(t), \dot{x}(t), t) - \mathcal{V}(x(t), \dot{x}(t), t), \quad (2.8)$$

gdzie:

- \mathcal{T} – energia kinetyczna,
- \mathcal{V} – energia potencjalna.

Aby odnaleźć ekstremalę funkcji kosztu (2.7), należy rozwiązać równanie Eulera - Lagrange'a:

$$\frac{d}{dt} \left(\frac{\partial L}{\partial \dot{x}} \right) - \frac{\partial L}{\partial x} = 0, \quad (2.9)$$

gdzie:

- $\frac{\partial L}{\partial \dot{x}}$ – pęd uogólniony,
- $\frac{\partial L}{\partial x}$ – siła uogólniona.

PRZYKŁAD 2.2

Oscylator harmoniczny (ciało na sprężynie). Dla oscylatora harmonicznego można wyprowadzić Lagrangian:

$$L = \frac{1}{2}m\dot{x}^2 - \frac{1}{2}kx^2, \quad (2.10)$$

gdzie:

- x – położenie,
- m – masa ciała na sprężynie,
- k – stała sprężystości sprężyny.

Podstawiając do (2.9) otrzymuje się:

$$\frac{d}{dt} (m\dot{x}) + kx = 0. \quad (2.11)$$

Ostatecznie po uproszczeniu:

$$\ddot{x} + \frac{k}{m}x = 0. \quad (2.12)$$

2.4. OPTYMALIZACJA LINIOWO-KWADRATOWA

Stabilność klasycznych układów można interpretować jako dążenie do minimalizacji funkcji energii układu, której charakterystyczną cechą jest ciągły spadek wartości. W przypadku, gdy energia układu autonomicznego jest wytracana aż do osiągnięcia punktu równowagi dynamicznej, układ uznawany jest za stabilny. W związku z powyższym, zadaniem optymalizacji jest minimalizacja kwadratowego funkcjonału kosztu:

$$\min J = \int_0^\infty (x^T Q x), \quad (2.13)$$

gdzie:

- $Q = \mathbb{I}$ – jednostkowa macierz kosztu,

- x – wektor stanu.

W normalnych warunkach rozwiązywanie zadania optymalizacji sprowadza się do rozwiązywania algebraicznego równania Riccatiego:

$$A^T P + PA - PBR^{-1}B^T P + Q = 0. \quad (2.14)$$

W rozważanym przypadku równanie upraszcza się:

$$A^T P + PA + Q = 0, \quad (2.15)$$

ponieważ cechą układów autonomicznych jest brak oddziaływanego z zewnętrznym sterowaniem, dzięki czemu części równania, które zawierają macierz wejść B są równe zeru.

Do rozwiązywania równania Riccatiego można wykorzystać gotowe funkcje środowiska Matlab [6]. Biblioteka Control System Toolbox zawiera dwie możliwości. Pierwszą z nich jest użycie funkcji `icare` zgodnie z dokumentacją [13], drugą natomiast jest wykorzystanie `lyap`; listing 2.1.

Listing 2.1: Funkcje icare i lyap

```

1 [X, K, L] = icare(A, B, Q, R, S, E, G)
2 X = lyap(A, Q)

```

Po rozwiązaniu zależności uzyskuje się macierz P , której forma kwadratowa:

$$\mathcal{V} = x^T Px, \quad (2.16)$$

będzie tworzyła hiperfunkcję przestrzeni stanu, będącą minimalną reprezentacją energii potencjalnej trajektorii układu.

PRZYKŁAD 2.3

Dany jest układ opisany macierzą stanu A_1 (2.4) z warunkiem początkowym równym $x_0 = [1 \ 1]^T$; bez wymuszenia. Zadanie polega na odnalezieniu funkcji energii układu.

Przy użyciu funkcji `lyap` można szybko odnaleźć taką funkcję. Otrzymane rezultaty są przedstawione poziomikami funkcji energii z Rysunku 2.2. Listing 2.2 jest przykładem wykorzystania `lyap` do obliczania formy kwadratowej macierzy P (2.16); funkcji energii potencjalnej układu. Początkowo ustawiana jest jednostkowa macierz kosztu Q o rozmiarze macierzy stanu. Następnie z `lyap` otrzymuje się rozwiązanie równania Riccatiego:

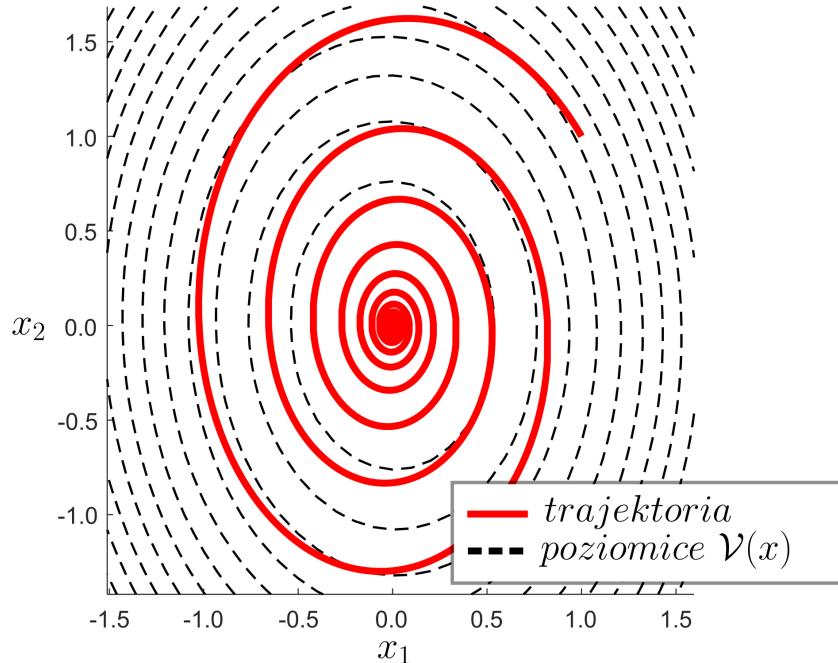
$$P = \begin{bmatrix} 7.487 & 0.124 \\ 0.124 & 3.756 \end{bmatrix}. \quad (2.17)$$

Określenie punktów siatki umożliwia dyskretyzację funkcji energii, natomiast pętla `for` oblicza wartości funkcji, dla przyjętych parametrów dyskretyzacji - krok 0.1 - iterując przez

współrzędne. Wynikiem działania programu jest uzyskanie wartości hiperfunkcji nad przestrzenią stanu - przestrzeń oprócz współrzędnych stanu, ma również nadane wartości w każdym punkcie. Określony zabieg definiuje się odwzorowaniem:

$$\mathcal{V} : x \in X \subseteq \mathbb{R}^n \quad \mathcal{V}(x) = x^T P x. \quad (2.18)$$

Wartości funkcji $\mathcal{V}(x(t))$ zostały wykreślone na Rysunku 2.3.



Rysunek 2.2: Trajektorie układu i poziomice funkcji energii

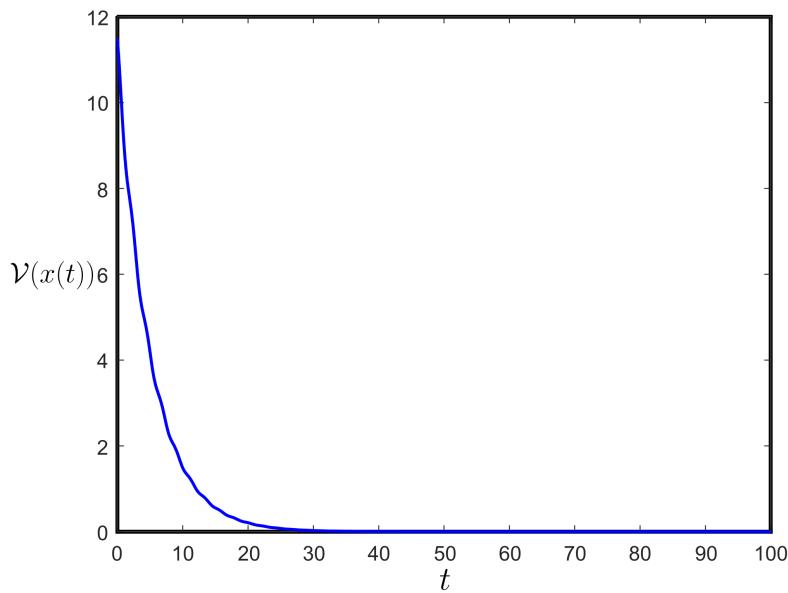
Listing 2.2: Obliczanie funkcji energii i jej wartości w przestrzeni stanu

```

1   Q = eye(size(A1)); % macierz Q
2   P = lyap(A1', Q); % rozwiazanie rownania A'*P + P*A = -Q
3
4   % Zakres i siatka dla poziomic
5   x_range = -2:0.1:2;
6   y_range = -2:0.1:2;
7   [X, Y] = meshgrid(x_range, y_range);
8
9   % Obliczenie wartosci funkcji
10  V = zeros(size(X));
11  for i = 1:numel(X)
12      z = [X(i); Y(i)];
13      V(i) = z' * P * z;
14  end

```

Analizy trajektorii jako naturalnych ekstremali oraz funkcji energii prowadzą do istotnych wniosków dotyczących dynamiki układów fizycznych. Okazuje się, że trajektorie nie tylko opisują zmiany wielkości w układach, ale również dostarczają informacji o ich stabilności. Układ, który zaczyna swój ruch z określonego stanu początkowego x_0 , porusza się po trajektorii spiralnej, jak pokazano na Rysunku 2.2, w układzie współrzędnych przekształconym przez macierz stanu. Z kolei odpowiednio dobrana funkcja energii, przedstawiona na Rysunku 2.3, wykazuje ciągły spadek, co jest oczekiwanym zachowaniem układu, na który nie działają żadne zewnętrzne siły. Proces ten wskazuje na naturalne wytracanie energii, aż do osiągnięcia zera, co jest dowodem stabilności badanego układu. Pod koniec XIX stulecia uczony Aleksandr Lapunow do-



Rysunek 2.3: Energia potencjalna układu

szedł do podobnych spostrzeżeń i sformalizował metodę badania stabilności, nazywaną dziś metodą Lapunowa. Dzięki tej metodzie możliwe jest określenie warunków, w których układ zachowuje stabilność. Zastosowanie funkcji energii jako narzędzia do analizy trajektorii w połączeniu z metodą Lapunowa pozwala na skuteczną ocenę stabilności układu oraz przewidywań jego zachowania w długim okresie czasu. Następne podrozdziały będą poświęcone formalnemu zapisowi wybranych i kluczowych dla pracy, kryteriów stabilności w sensie Lapunowa. Rozwinięcie zagadnień pozwoli lepiej zrozumieć sens stosowania metody i jej ograniczenia.

2.5. BEZPOŚREDNIA METODA LAPUNOWA

Rozwiążanie równania stanu (2.2) zapisuje się również jako funkcje zależne od warunków początkowych i czasu:

$$\mathbf{x}(t) = \mathbf{p}(t, \mathbf{x}_0, t_0). \quad (2.19)$$

Punkt równowagi dynamicznej układu to rozwiązanie równania stanu dla zerowej pochodnej czasowej, czyli stanu, w którym nie zachodzą żadne zmiany wektora stanu:

$$x_i^e : \quad \mathbf{0} = \sum_{j=1}^n a_{ij}x_j(t) + b_iu(t), \quad i = 1, 2, \dots, n, \quad (2.20)$$

gdzie:

- $\mathbf{x}^e = [x_1^e, x_2^e, \dots, x_n^e]^T \in X \subseteq \mathbb{R}^n$ – punkt równowagi dynamicznej.

Atraktorem w sensie Lapunowa nazywamy punkt równowagi dynamicznej lub zbiór stanów, który spełnia warunek:

$$\forall |\mathbf{x}_0 - \mathbf{x}_A| < \eta, \quad \lim_{t \rightarrow \infty} \mathbf{p}(t, \mathbf{x}_0, t_0) \rightarrow \mathcal{A}, \quad \eta > 0, \quad (2.21)$$

gdzie:

- $\mathbf{x}_A \in \mathcal{A} \subseteq X \subseteq \mathbb{R}^n$ – atraktor,
- $|\mathbf{x}_0 - \mathbf{x}_A|$ – promień otoczenia atraktora,
- \mathcal{A} – zbiór atraktorów układu.

Oznacza to, że dla liczby $\eta > 0$, istnieje takie otoczenie atraktora, w którym odległość warunku początkowego od tego atraktora jest mniejsza niż η . Jeśli dodatkowo trajektoria układu zbiega do atraktora wraz z upływem czasu ($t \rightarrow \infty$), to taki punkt lub zbiór można uznać za atraktor.

PRZYKŁAD 2.4

Klasycznym przykładem systemu nieliniowego z chaotycznym atraktorem jest układ Lorentza:

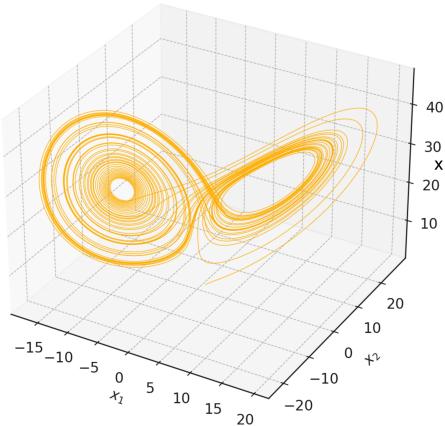
$$\begin{cases} \dot{x}_1 = \sigma(x_2 - x_1), \\ \dot{x}_2 = x_1(\rho - x_3) - x_2, \\ \dot{x}_3 = x_1x_2 - \beta x_3, \end{cases} \quad (2.22)$$

Atraktor Lorentza charakteryzuje się strukturą przyciągającą trajektorie układu, jednak w przeciwieństwie do stabilnego punktu równowagi nie prowadzi on do zbieżności trajektorii do jednej ustalonej wartości. Trajektorie oscylują między dwoma obszarami przestrzeni stanu w sposób nieregularny - Rysunek 2.4 - tworząc spójny, chaotyczny obiekt matematyczny. Układ ma tylko jeden atraktor, ponieważ opisywany jest jednym układem równań. Listing 2.3, przedstawia w pełni funkcjonujący skrypt, który generuje trajektorie układu Lorentza w przestrzeni stanu.

Listing 2.3: Skrypt w Python, generujący trajektorię układu Lorentza

```
1 import numpy as np
2 from scipy.integrate import solve_ivp
3 import matplotlib.pyplot as plt
4
5 # Parametry
6 sigma = 10.0
7 rho = 28.0
8 beta = 8.0 / 3.0
9
10 # Równania
11 def lorenz(t, state):
12     x, y, z = state
13     dxdt = sigma * (y - x)
14     dydt = x * (rho - z) - y
15     dzdt = x * y - beta * z
16     return [dxdt, dydt, dzdt]
17
18 # Warunki początkowe i czas symulacji
19 initial_state = [1.0, 1.0, 1.0]
20 t_span = (0, 50)
21 t_eval = np.linspace(t_span[0], t_span[1], 10000)
22
23 # Rozwiazanie
24 solution = solve_ivp(lorenz, t_span, initial_state, t_eval=t_eval)
25
26 # Wykres
27 fig = plt.figure(figsize=(10, 7))
28 ax = fig.add_subplot(projection='3d')
29 ax.plot(solution.y[0], solution.y[1], solution.y[2], lw=0.5)
30 ax.set_xlabel("$x_1$")
31 ax.set_ylabel("$x_2$")
32 ax.set_zlabel("$x_3$")
33 plt.show()
```

Obliczenia numeryczne trajektorii w środowisku języka Python [9] są możliwe dzięki bibliotece `scipy`, która zawiera funkcje np. `solve_ivp` do rozwiązywania równań różniczkowych [8]. Biblioteka `numpy` [7] ułatwia wykonywanie obliczeń, natomiast dzięki `matplotlib` możliwe jest tworzenie wykresów [10].

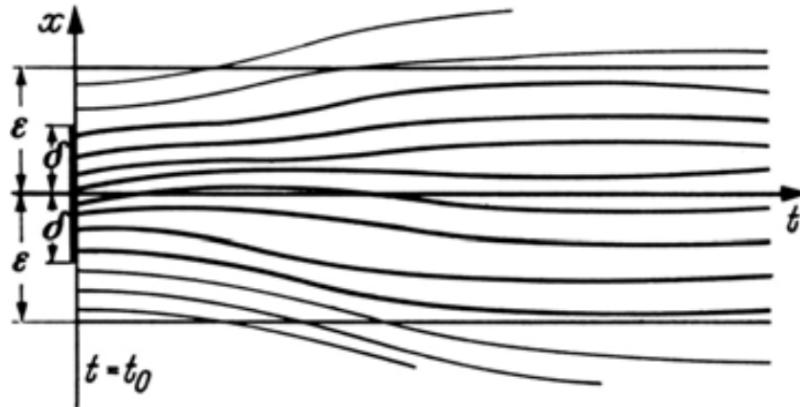


Rysunek 2.4: Trajektoria układu Lorentza

Trajektoria jest stabilna, jeśli istnieje taka odległość δ od punktu równowagi dynamicznej, że dla każdego punktu początkowego znajdującego się w tej odległości, trajektoria układu nigdy nie oddali się dalej niż o ε od punktu równowagi, niezależnie od tego, jak długo obserwuje się zachowanie układu:

$$\forall \varepsilon > 0 \exists \delta > 0 : |\mathbf{x}_0| < \delta \implies \lim_{t \rightarrow \infty} |\mathbf{p}(t, \mathbf{x}_0, t_0)| < \varepsilon. \quad (2.23)$$

Możliwe trajektorie przedstawia Rysunek 2.5, na którym można zaobserwować układy, które spełniają i niespełniają warunku (2.23).



Rysunek 2.5: Trajektorie w płaszczyźnie (x, t) . Źródło: [18], str. 6

Konsekwencją poprzednich rozważań jest wypracowanie definicji na stabilność asymptotyczną układu.

DEFINICJA 2.1

System jest stabilny asymptotycznie jeśli jest stabilny w sensie Lapunowa (2.23) i można w nim określić przynajmniej jeden atraktor (2.21).

Na podstawie rysunku 2.2 oraz warunku stabilności w sensie Lapunowa (2.23) można zauważać istotną zależność. Jeżeli warunek początkowy spełnia $|x_0| < \delta$, to istnieje nieskończonie wiele wartości $\varepsilon < \delta$, które wyznaczają odległości między punktem początkowym a punktem równowagi dynamicznej (pełniącym w tym przypadku również rolę atraktora). Trajektoria układu konsekwentnie przechodzi przez te wartości ε , co można interpretować jako sukcesywne zbliżanie się do punktu równowagi. Aby zbadać stabilność takiego układu, wystarczy znaleźć dodatnio określoną funkcję energii $\mathcal{V}(x)$, której poziomice odpowiadają wartościom ε , a następnie sprawdzić, czy trajektoria układu systematycznie zmniejsza wartość ε w czasie. Wymienioną metodę badania stabilności określa się bezpośrednią metodą Lapunowa:

$$\forall t \geq t_0, \dot{\mathcal{V}}(x(t)) < 0 \quad \wedge \quad \mathcal{V}(x(t)) = \varepsilon. \quad (2.24)$$

Warto zauważyć, że $|x_0|$ nie jest odlegością w sensie euklidesowym. Wynika to z faktu, że macierz stanu przekształca przestrzeń stanu w sposób, który powoduje deformację okręgów – zamiast zachować ich klasyczny, euklidesowy kształt, przyjmują one formę elips lub innych krzywych, w zależności od charakterystyki przekształcenia. W konsekwencji normy w przestrzeni stanu nie mogą być interpretowane w klasyczny sposób. Do określenia kształtu przekształconej przestrzeni można zastosować wspomnianą wcześniej metodę optymalizacji, która dopasuje hiperfunkcję do struktury przekształceń. Na tej podstawie można wyznaczyć nie tylko równe odległości ε od punktu równowagi, ale również normy w tej przestrzeni. Wartość funkcji $\mathcal{V}(x)$ będzie stanowiła miarę normy odległości punktu od początku przestrzeni.

2.6. PODSUMOWANIE ROZDZIAŁU

W niniejszym rozdziale omówiono podstawy badania stabilności układów dynamicznych w sensie Lapunowa, przedstawiając głównie klasyczne modele liniowe i stacjonarne. Metoda Lapunowa umożliwia ocenę stabilności układów na podstawie analizy funkcji energii. Pokazano, że modelowanie może być skutecznie realizowane przy użyciu formalizmu mechaniki Newtonowskiej oraz zasady Hamiltona. Rozdział uwypuklił znaczenie funkcji Lapunowa jako narzędzia analitycznego, które pozwala nie tylko ocenić stabilność układu, ale także zrozumieć jego zachowanie w długim horyzoncie czasowym. Omówiono pojęcia punktów równowagi i atraktorów, które odgrywają znaczącą rolę w zrozumieniu dynamiki systemów. Przedstawiono także podejście optymalizacyjne, oparte na minimalizacji jednostkowego, kwadratowego funkcjonału kosztu, które może być stosowane do obliczania parametrów kwadratowej funkcji energii. Podkreślono, że równanie Riccatiego, jest efektywnie obliczane w środowiskach obliczeniowych, co jest szczególnie istotne w praktyce inżynierskiej. Podsumowując, zaprezentowane w rozdziale metody i przykłady potwierdzają, że podejście oparte na metodzie Lapunowa stanowi podstawę do analizy stabilności układów dynamicznych, zarówno w kontekście teoretycznym, jak i praktycznym. Wyniki i obserwacje są punktem wyjścia do dalszych badań, szczególnie w odniesieniu do bardziej złożonych układów o niecałkowitym rzędzie.

3. ANALIZA STABILNOŚCI UKŁADÓW NIECAŁKOWITEGO RZĘDU

3.1. WPROWADZENIE

Bieżący rozdział zostanie poświęcony podstawowym definicjom różniczek i całek ułamkowego rzędu, ze szczególnym uwzględnieniem podejść: Riemann-Liouville'a, Caputo oraz Grünwalda-Letnikowa. Każda z tych definicji znajduje zastosowanie w różnych kontekstach i posiada unikalne właściwości, które zostaną omówione. Ponadto zostaną zaprezentowane metody analizy numerycznej w środowisku MATLAB, umożliwiające praktyczną implementację tych definicji. Szczególna uwaga zostanie poświęcona stabilności układów dynamicznych o niecałkowitym rzędzie. Następnie zostaną przedstawione analizy autora pracy, które dotyczą stabilności, które uwzględniają specyficzne cechy rachunku, takie jak pamięć układu czy wpływ zmiennych warunków początkowych. Rozważone zostaną również trajektorie układów w przestrzeni stanu oraz ich wpływ na funkcje Lapunowa. Na koniec zostaną zaprezentowane trzy metody doboru funkcji Lapunowa do dynamiki układu o rzędzie niecałkowitym. Pierwsza z nich polega na zauważeniu, że funkcja Lapunowa układu całkowitego rzędu zachowuje swoje warunki dla układów o mniejszych rzędach. Druga metoda opiera się na minimalizacji funkcjonału kosztu, przy wykorzystaniu algorytmu simpleks. Ostatnia metoda polega na optymalizacji funkcji do granicznego spełnienia warunku zachowania ujemności pochodnej.

3.2. DEFINICJE RACHUNKU NIECAŁKOWITEGO RZĘDU

W rachunku różniczkowo-całkowym rzędu ułamkowego można wyróżnić kilka różnych definicji, dostosowanych do specyfiki i potrzeb modeli matematycznych. Jedną z najczęściej stosowanych i bardziej popularnych definicji całkowania ułamkowego rzędu jest definicja według Riemann-Liouville'a [28]:

$${}_0I_t^\alpha f(t) = \frac{1}{\Gamma(\alpha)} \int_0^t (t-\tau)^{\alpha-1} f(\tau) d\tau, \quad (3.1)$$

gdzie:

- $\Gamma(\alpha) = \int_0^\infty t^{\alpha-1} e^{-t} dt$, $\mathbb{R}(\alpha) > 0$ – funkcja Gamma,
- $\alpha \in \mathbb{R}^+$ – rząd całkowania.

Definicja pochodnej w sensie Riemann-Liouville'a opiera się na wcześniej przedstawionej całce ułamkowego rzędu (3.1), na której stosuje się różniczkowanie całkowitego rzędu:

$$\begin{aligned} {}_{0}^{RL}\mathcal{D}_t^{\alpha} f(t) &= \frac{d^{\alpha}}{dt^{\alpha}} f(t) = \frac{d^N}{dt^N} [{}_{0}I_t^{N-\alpha} f(t)] = \\ &= \frac{1}{\Gamma(N-\alpha)} \frac{d^N}{dt^N} \int_0^t (t-\tau)^{N-\alpha-1} f(\tau) d\tau, \end{aligned} \quad (3.2)$$

gdzie:

- $N - 1 < \alpha < N \in \mathbb{N}$.

W wyniku tego procesu uzyskuje się pochodne rzędu ułamkowego. Warto zauważyć, że w standardowych warunkach rozwiązaniem operacji całkowania jest funkcja pierwotna oraz stała całkowa. Zastosowanie różniczkowania na takim rozwiązaniu eliminuje stałą całkową, co powoduje, że rachunek Riemanna-Liouville'a nie uwzględnia warunków początkowych w klasycznym sensie.

Drugą popularną definicją jest różniczka w sensie Caputo:

$${}_{0}^C\mathcal{D}_t^{\alpha} f(t) = \frac{1}{\Gamma(N-\alpha)} \int_0^t \frac{f^N(\tau)}{(t-\tau)^{\alpha+1-N}} d\tau, \quad (3.3)$$

W tym przypadku podejście różni się od definicji Riemanna-Liouville'a, ponieważ pochodna czasowa znajduje się wewnątrz całki. Dzięki temu warunki początkowe mogą być zachowane w ich klasycznej postaci. Taka forma różniczkowania na pozór wydaje się szczególnie użyteczna w zastosowaniach fizyki, zwłaszcza w badaniach stabilności układów z wykorzystaniem metody Lapunowa.

Istotnym narzędziem potrzebnym do wykonywania różnorodnych operacji w rachunku niecałkowitego rzędu jest transformata Laplace'a:

$$\mathcal{L}\{\mathcal{D}^{\alpha}(t)\}(s) = s^{\alpha} F(s) - \sum_{k=1}^{\lceil \alpha \rceil} (s^{\alpha-k}) \left(\mathcal{D}^{k-1} \Big|_{t=0} \right). \quad (3.4)$$

W wypadku definicji Riemanna-Liouville'a, transformata Laplace'a przybiera postać:

$$\mathcal{L}\{{}_{0}^{RL}\mathcal{D}_t^{\alpha} f\}(s) = s^{\alpha} F(s) - \sum_{k=1}^{\lceil \alpha \rceil} (s^{k-1}) \left({}_{0}^{RL}\mathcal{D}_t^{\alpha-k} \Big|_{t=0} \right), \quad (3.5)$$

natomiast w wypadku definicji Caputo transformata Laplace'a jest:

$$\mathcal{L}\{{}_{0}^C\mathcal{D}_t^{\alpha} f\}(s) = s^{\alpha} F(s) - \sum_{k=1}^{\lceil \alpha \rceil} (s^{\alpha-k}) \left({}_{0}^C\mathcal{D}_t^{\alpha-k} \Big|_{t=0} \right). \quad (3.6)$$

Trzecią definicją jest różniczkowanie dyskretnie w sensie Grünwalda–Letnikova [19]:

$${}_0\Delta_k^\alpha x_i = \sum_{j=0}^k (-1)^j \binom{\alpha}{j} x_{i-j}, \quad (3.7)$$

gdzie:

- $0 < \alpha < 1$, $\alpha \in \mathbb{R}$,
- x_i – funkcja dyskretna,

$$\binom{\alpha}{j} = \begin{cases} 1, & \text{for } j = 0, \\ \frac{\alpha(\alpha-1)\cdots(\alpha-j+1)}{j!}, & \text{for } j = 1, 2, 3, \dots \end{cases}$$

która umożliwia odtworzenie rachunku różniczkowego ułamkowych rzędów, w dziedzinie czasu dyskretnego.

Istnieje wiele definicji rachunku różniczkowo-całkowego ułamkowych rzędów, jednak w niniejszej pracy analizie poddane zostaną wyłącznie trzy podstawowe definicje, ze względu na ich szerokie zastosowanie i popularność.

3.3. REPREZENTACJE NUMERYCZNE W MATLAB

Środowisko Simulink [14], dostępne w programie Matlab, oferuje szereg bibliotek opracowanych przez ekspertów w dziedzinie rachunku różniczkowego i całkowego rzędów ułamkowych. Jedną z najbardziej szczegółowo opisanych bibliotek jest FOMCON (Fractional-order Modeling and Control) [15]. Biblioteka ta opiera się na zmodyfikowanej wersji aproksymacji Oustaloupa [27, 38], która uwzględnia cechy charakterystyczne dla transmitancji operatorowej; brak warunków początkowych i wykorzystuje transformację Laplace'a rachunku niecałkowitego rzędu:

$$s^\alpha \approx G_\omega \prod_{k=-N}^N \frac{s + \omega'_k}{s + \omega_k} \quad (3.8)$$

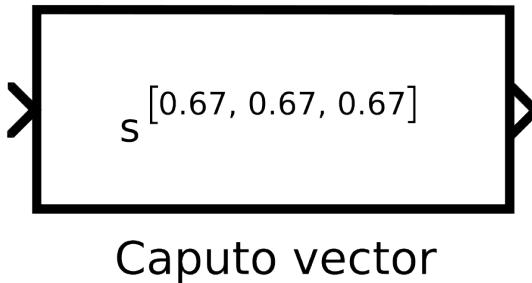
gdzie:

- $G_\omega = \left(\frac{d\omega_h}{b}\right)^\alpha \left(\frac{ds^2 + b\omega_h s}{d(1-\alpha)s^2 + b\omega_h s + d\alpha}\right)$,
- $\omega_k = \left(\frac{b\omega_h}{d}\right)^{\frac{\alpha+2k}{2N+1}}$, $\omega'_k = \left(\frac{d\omega_b}{b}\right)^{\frac{\alpha-2k}{2N+1}}$,

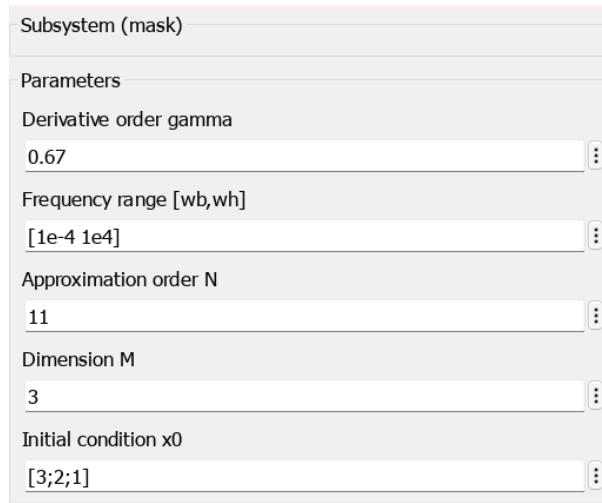
Aproksymacja (3.8) umożliwia modelowanie rzędów ułamkowych w wyznaczonym paśmie częstotliwości $\omega = (\omega_b, \omega_h)$, przy czym dokładność odwzorowania zależy również od przyjętej liczby składników szeregu N , natomiast $b > 0$, $d > 0$.

Jednym z ważnych narzędzi wspierających analizę układów o niecałkowitych rzędach jest FOTF Toolbox [4, 11]. Biblioteka ta umożliwia numeryczne rozwiązywanie problemów rachunku różniczkowo-całkowego rzędów ułamkowych, wykorzystując blok operatora Caputo

programu Simulink – przedstawiony na Rysunku 3.1. Zastosowany interfejs bloku (Rysunek 3.2) pozwala na precyzyjne określenie kluczowych parametrów, takich jak rząd operatora różniczkowania, zakres pasma częstotliwości stosowanego w aproksymacji, poziom dokładności aproksymacji, rząd macierzy opisujących układ oraz warunki początkowe.



Rysunek 3.1: Blok Caputo Vector w Simulink



Rysunek 3.2: Maska bloku Caputo Vector

3.4. NUMERYCZNA I ANALITYCZNA POCHODNA FUNKCJI HEAVISIDE'A

Klasyczna teoria sygnałów zakłada, że dowolną funkcję ciągłą można przybliżyć jej dyskretną reprezentacją, przy zastosowaniu odpowiedniego okresu próbkowania. W tej sekcji przeanalizowano możliwość zastosowania tego podejścia w przypadku rachunku różniczkowo-całkowego rzędów ułamkowych, wykorzystując pochodną funkcji skoku jednostkowego jako przykład do analizy. Według definicji Riemann-Liouville'a pochodną funkcji skoku jednostkowego:

$$f(t) = \mathbb{1}(t) = \begin{cases} 0, & t < 0, \\ 1, & t \geq 0, \end{cases} \quad (3.9)$$

można wyprowadzić w następujący sposób:

$$\begin{aligned}
 \frac{d^\alpha}{dt^\alpha} \mathbb{1}(t) &= \frac{1}{\Gamma(N-\alpha)} \frac{d^N}{dt^N} \int_0^t (t-\tau)^{N-\alpha-1} d\tau \\
 &= \frac{1}{\Gamma(N-\alpha)} \frac{d^N}{dt^N} \left[\frac{-1}{N-\alpha} (t-\tau)^{N-\alpha} \right]_0^t \\
 &= \frac{1}{\Gamma(N-\alpha)} \frac{1}{N-\alpha} \frac{d^N}{dt^N} t^{N-\alpha} \\
 &= \frac{1}{\Gamma(N-\alpha)} \frac{1}{N-\alpha} (N-\alpha)(N-\alpha-1) \cdots (1-\alpha) t^{-\alpha} \\
 &= \frac{t^{-\alpha}}{\Gamma(1-\alpha)}.
 \end{aligned} \tag{3.10}$$

Pochodna dla przykładowego rzędu $\alpha = 0.5$ jest następująca:

$$\frac{d^{0.5}}{dt^{0.5}} \mathbb{1}(t) = \frac{t^{-0.5}}{\Gamma(0.5)}. \tag{3.11}$$

Definicja Grünwalda–Letnikova (3.7) dla rzędu $\alpha = 0.5$ przyjmuje postać:

$$\begin{aligned}
 0. \quad x_0 &= (-1)^0 \binom{\alpha}{j} = 1, \quad \text{dla } j = 0, k = 0 \\
 \Delta^\alpha x_0 &= 1 \\
 1. \quad (-1)^0 \binom{\alpha}{j} x_1 &= x_1 \quad \text{dla } j = 0, k = 1 \\
 (-1) \cdot 0.5 \cdot x_0 &= (-0.5) \quad \text{dla } j = 1, k = 1 \\
 \Delta^\alpha x_1 &= x_1 - \frac{1}{2} \\
 2. \quad (-1)^0 \binom{\alpha}{j} x_2 &= x_2 \quad \text{dla } j = 0, k = 2 \\
 (-1) \cdot 0.5 \cdot x_1 &= -0.5x_1 \quad \text{dla } j = 1, k = 2 \\
 \frac{\alpha(\alpha-1)}{2} x_0 &= \frac{0.5(-0.5)}{2} x_0 = -0.125x_0 \quad \text{dla } j = 2, k = 2 \\
 \Delta^\alpha x_2 &= x_2 - \frac{1}{2}x_1 - \frac{1}{8}x_0
 \end{aligned} \tag{3.12}$$

Z tego otrzymuje się kolejne wyrażenia na pochodne niecałkowitego rzędu zgodne z definicją (3.7), dla jednostkowego okresu próbkowania:

$$\begin{aligned}
 \Delta^\alpha x_0 &= 1 \\
 \Delta^\alpha x_1 &= x_1 - \frac{1}{2}x_0 \\
 \Delta^\alpha x_2 &= x_2 - \frac{1}{2}x_1 - \frac{1}{8}x_0
 \end{aligned} \tag{3.13}$$

$$\Delta^\alpha x_3 = x_3 - \frac{1}{2}x_2 - \frac{1}{8}x_1 - \frac{1}{16}x_0$$

...

Ręczne wyprowadzenia umożliwiają odtworzenie funkcji do obliczeń numerycznych definicji Grünwalda–Letnikova i porównanie jej z ciągłą pochodną skoku jednostkowego - Listing 3.1.

Listing 3.1: Kod do obliczania i porównania definicji ciągłych i dyskretnych

```

1 alpha2 = 0.5;
2
3 A = 0;
4 DIM = 1;
5 T = 0.015;
6 dt = 0.00001;
7 k = T/dt;
8 t = linspace(0, T, k);
9
10 %% G1
11 for i=1:length(t)
12     Df(i) = sum(back_v(alpha2,i))/dt;
13 end
14
15 for i=1:length(t)
16     Df2(i) = sum(back_v(alpha,i))/dt;
17 end
18 %% RL
19 for i=1:length(t)
20     RL(1,i) = (i^(-alpha2))/gamma(1-alpha2)/dt;
21 end
22 for i=1:length(t)
23     RL2(1,i) = (i^(-alpha))/gamma(1-alpha)/dt;
24 end

```

Funkcja `back_v` jest pomocnicza i oblicza wektor wstecznych współczynników $\binom{\alpha}{j}$ - Listing 3.2.

Listing 3.2: Funkcja `back_v`

```

1 for j=1:i-1
2     a_v(j+1) = a_v(j)*(1-(1+alpha)/(j));
3 end

```

Do dalszej analizy pozostała jedynie definicja Caputo (3.3). Wewnątrz definicji znajduje się pochodna funkcji skoku jednostkowego, czyli delta Diracka $\delta(t)$:

$${}_0^C\mathcal{D}_t^\alpha \mathbb{1}(t) = \frac{1}{N - \alpha} \int_0^t \frac{\delta(\tau)}{(t - \tau)^{\alpha+1-N}} d\tau. \quad (3.14)$$

W takim wypadku trzeba obliczyć przedstawioną całkę splotu. Przykładowo można użyć metody całkowania przez części:

$$\int \delta(t)(t - \tau)^{N-\alpha-1} d\tau = f(\tau)g(\tau) - \int f(\tau)g'(\tau). \quad (3.15)$$

Do obliczenia pochodnej funkcji $g(\tau)$ wykorzystano regułę łańcuchową:

$$\begin{aligned} u &= t - \tau, \quad \frac{d(t - \tau)^{N-\alpha-1}}{d\tau} = \frac{du^{N-\alpha-1}}{du} \cdot \frac{du}{d\tau} \\ &= (N - \alpha - 1)u^{N-\alpha-2}(-1) = (\alpha - N + 1)(t - \tau)^{N-\alpha-2}. \end{aligned} \quad (3.16)$$

Znając wszystkie składniki całkowania przez części:

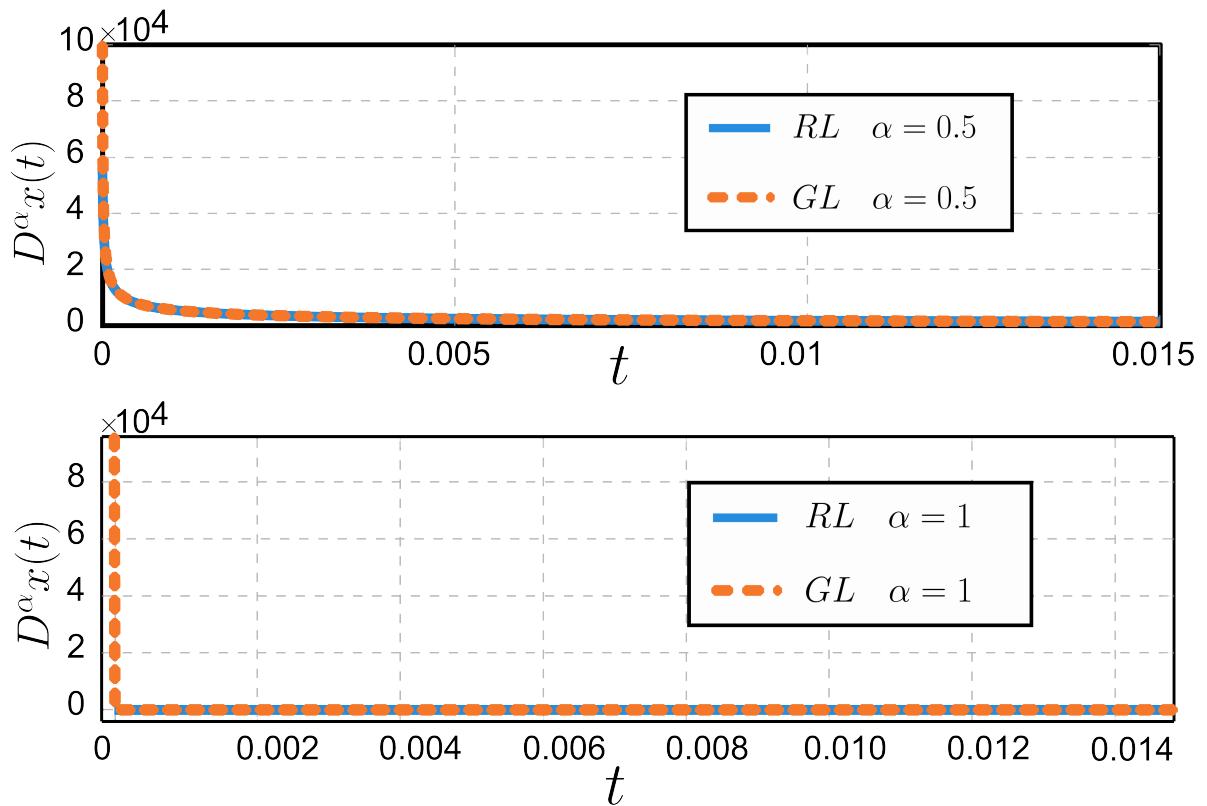
$$f'(\tau) = \delta(\tau), \quad f(\tau) = 1, \quad g = (t - \tau)^{N-\alpha-1}, \quad g' = (\alpha - N + 1)(t - \tau)^{N-\alpha-2}, \quad (3.17)$$

można przystąpić do podstawiania i dalszych wyprowadzeń:

$$\begin{aligned} \int \delta(t)(t - \tau)^{N-\alpha-1} d\tau &= (t - \tau)^{N-\alpha-1} - \int (\alpha - N + 1)(t - \tau)^{N-\alpha-2} d\tau \\ &= (t - \tau)^{N-\alpha-1} - (\alpha - N + 1) \int (t - \tau)^{N-\alpha-2} d\tau \\ &= (t - \tau)^{N-\alpha-1} - (t - \tau)^{N-\alpha-1} + C(\tau) = 0 + C(\tau), \end{aligned} \quad (3.18)$$

które jednoznacznie dowodzą, że niezależnie od granic całkowania, w reakcji na funkcję skoku jednostkowego definicja Caputo będzie równa zeru lub stałej wartości, zależnie od warunku początkowego i innych założeń.

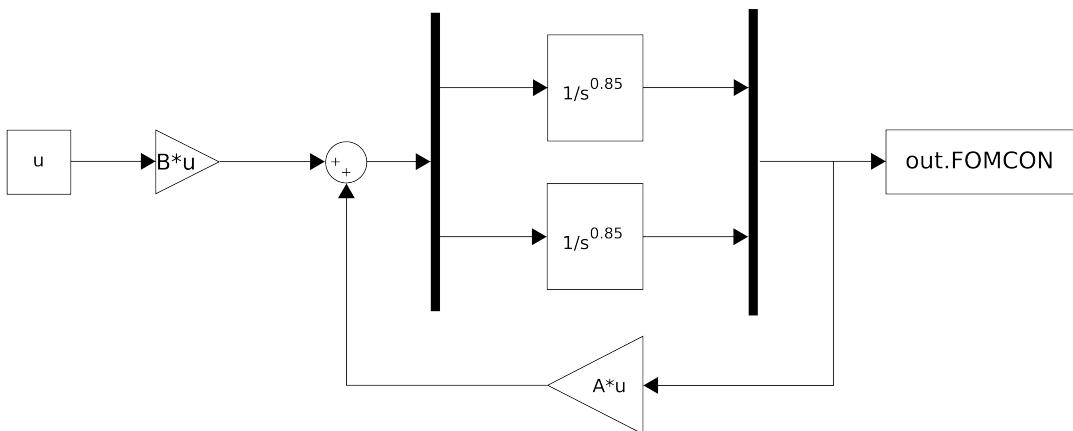
Rysunek 3.3 ilustruje, że dla tego samego okresu próbkowania definicje zachowują się podobnie zarówno w przypadku rzędów całkowitych, jak i niecałkowitych. Interesującą cechą rzędów niecałkowitych jest wpływ współczynników pamięci, które powodują, że funkcja nie osiąga zera natychmiast po pierwszym kroku, lecz zbliża się do niego stopniowo, w wydłużonym czasie. Takie zachowanie może prowadzić do niepożądanych efektów w układach z pamięcią, szczególnie podczas zadawania warunków początkowych, co wynika z charakterystyki rachunku różniczkowo-całkowego rzędów ułamkowych.



Rysunek 3.3: Porównanie analitycznej i dyskretnej pochodnej funkcji skoku jednostkowego

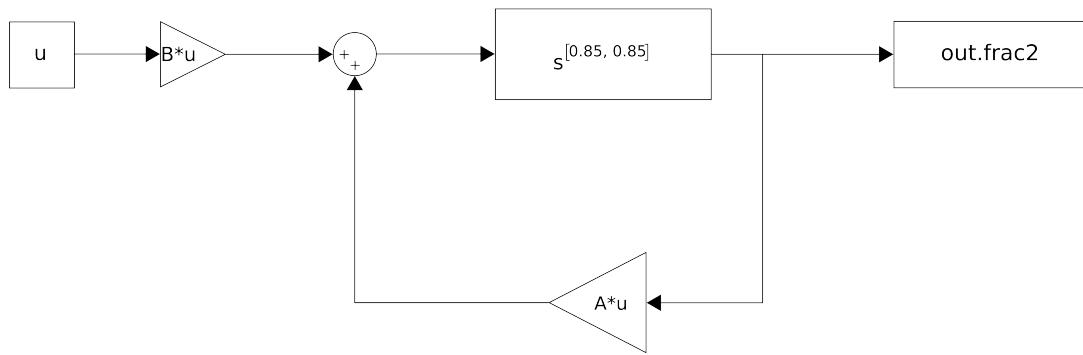
3.5. WARUNKI POCZĄTKOWE

W celu analizy zachowania rozwiązań numerycznych układów dynamicznych przy zmiennych warunkach początkowych zostaną wykorzystane modele układów drugiego rzędu w reprezentacji przestrzeni stanu. Analiza obejmuje układy o ułamkowym rzędzie $\alpha = 0.85$. Celem eksperymentu jest zbadanie wpływu różnych warunków początkowych na trajektorie układu oraz identyfikacja potencjalnych różnic między nimi. Macierz stanu jest zdefiniowana jako (2.4), macierz wejść ma postać $B = [0 \ 1]^T$, natomiast sterowanie wynosi $u = \mathbb{1}(t)$.



Rysunek 3.4: Układ drugiego rzędu, przy użyciu biblioteki FOMCON

Na Rysunku 3.4 przedstawiono reprezentację układu drugiego rzędu w bibliotece FOMCON. Ze względu na stosowany wariant aproksymacji Oustaloupa, w modelu tym nie można zadać warunków początkowych. Jednak podczas porównania różnych modeli, przy zerowych warunkach początkowych, dodatkowy model może posłużyć jako potwierdzenie zgodności reprezentacji numerycznych.



Rysunek 3.5: Układ drugiego rzędu w bibliotece FOTF

Układ drugiego rzędu z biblioteki FOTF przedstawia Rysunek 3.5. Można w nim zadać warunki początkowe, dlatego będzie odpowiednim modelem porównawczym do napisanej reprezentacji numerycznej definicji Grünwalda–Letnikova - Listing 3.3.

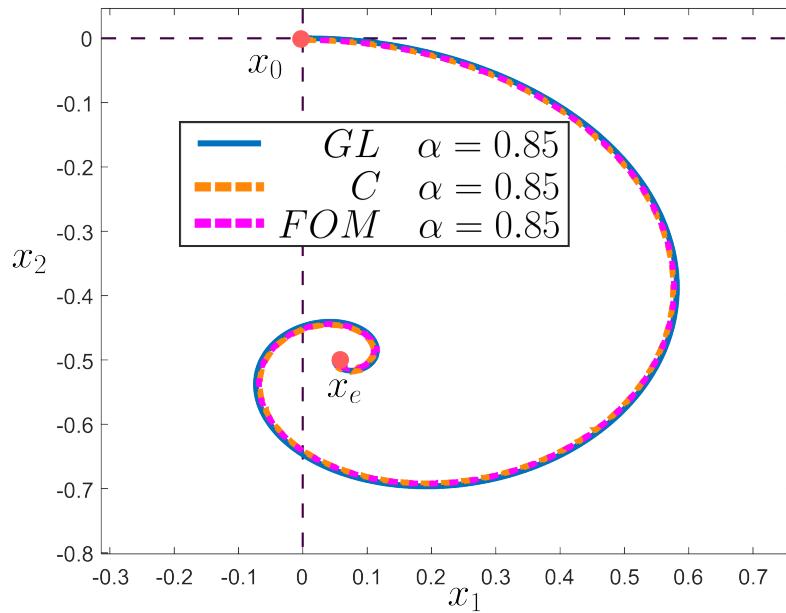
Listing 3.3: Model Grünwalda–Letnikova

```

1 %% petla glowna
2 for i=2:k
3     % a_v; wspolczynniki pamieci
4     a_v = back_v(alpha, i);
5
6     % otrzymanie wspolczynnikow pamieci
7     mem_coeff = a_v(2:end);
8
9     % poprzednie X
10    last_x = flip(X(:,1:i-1), 2);
11
12    % suma pamieci
13    mem_sum = sum(mem_coeff .* last_x, 2);
14
15    % obliczanie trajektorii
16    X(:,i) = A*X(:,i-1)*dt + (B * u)*dt - mem_sum;
17 end
  
```

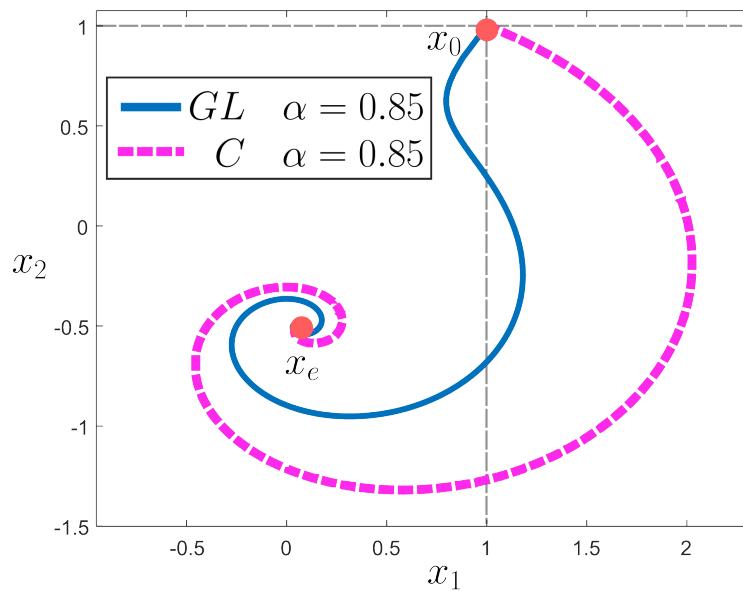
Trajektorie różnych reprezentacji numerycznych, uzyskane przy zerowych warunkach początkowych, zostały zestawione na Rysunku 3.6. Oznaczenia w legendzie odpowiadają następują-

cym modelom: GL - Grünwald–Letnikov, FOM - model biblioteki FOMCON, C - model FOTF. Analizując trajektorie wychodzące z zerowych warunków początkowych, można stwierdzić, że reprezentacje te są ze sobą zgodne i nie wykazują znaczących różnic.



Rysunek 3.6: Trajektorie układów przy zerowych warunkach początkowych

Rysunek 3.7 przedstawia porównanie trajektorii uzyskanych metodami Grünwald–Letnikov (GL) oraz FOTF (C) przy niezerowych warunkach początkowych. W tym przypadku trajektorie nie pokrywają się, co wskazuje na istotne różnice wynikające z przyjęcia niezerowych warunków początkowych. Różnice te sugerują, że zastosowanie bezpośredniej metody Lapunowa do analizy stabilności układów tej klasy może być problematyczne lub niemożliwe.



Rysunek 3.7: Trajektorie układów z niezerowych warunków początkowych

Jednocześnie można wskazać metodę, która umożliwi poprawę działania bezpośredniej metody Lapunowa. Zauważając, że w przypadku wymuszenia, przy zerowych warunkach początkowych, trajektorie różnych definicji pokrywają się, a układ dąży do osiągnięcia tego samego punktu równowagi dynamicznej, możliwe jest przesunięcie funkcji Lapunowa do x^e . Taki zbięg pozwala określić stabilność układu na podstawie przesuniętej funkcji energii układu. Jeśli układ pozostanie stabilny w zakresie działania wymuszenia, będzie to oznaczać jego stabilność w analizowanym zakresie przestrzeni sterowań.

3.6. TRANSLACJA W PRZESTRZENI STANU

Pierwszym krokiem niezbędnym do wykonania translacji jest określenie punktu równowagi dynamicznej układu. W tym celu należy rozwiązać równanie (2.20) dla zadanej macierzy stanu (2.4), wektora wymuszeń $B = [0, 1]^T$ oraz sterowania $u = \mathbb{1}(t)$. Po wprowadzeniu danych otrzymuje się:

$$\begin{bmatrix} 0 \\ 0 \end{bmatrix} = \begin{bmatrix} -0.1 & -1 \\ 2 & -0.1 \end{bmatrix} \begin{bmatrix} x_1^e \\ x_2^e \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \end{bmatrix} \quad (3.19)$$

Nawet w wypadku rachunku niecałkowitego rzędu równanie (3.19), będzie spełnione, ponieważ współczynniki pamięci definicji Grünwalda–Letnikova zanikają z czasem, natomiast każda następna próbka rozwiązania jest sumą poprzednich stanów układu i bez wymuszenia, w nieskończonym horyzoncie czasowym pamięć asymptotycznie zanika do zera, co jest zgodne z fizycznym rozumieniem modeli i paradoksem zbieżności. Wynikiem trywialnej równości jest wektor, który będzie wektorem translacji przestrzeni stanu:

$$\mathbf{v} = \begin{bmatrix} x_1^e \\ x_2^e \end{bmatrix} = \begin{bmatrix} 0.5025 \\ -0.05025 \end{bmatrix} \quad (3.20)$$

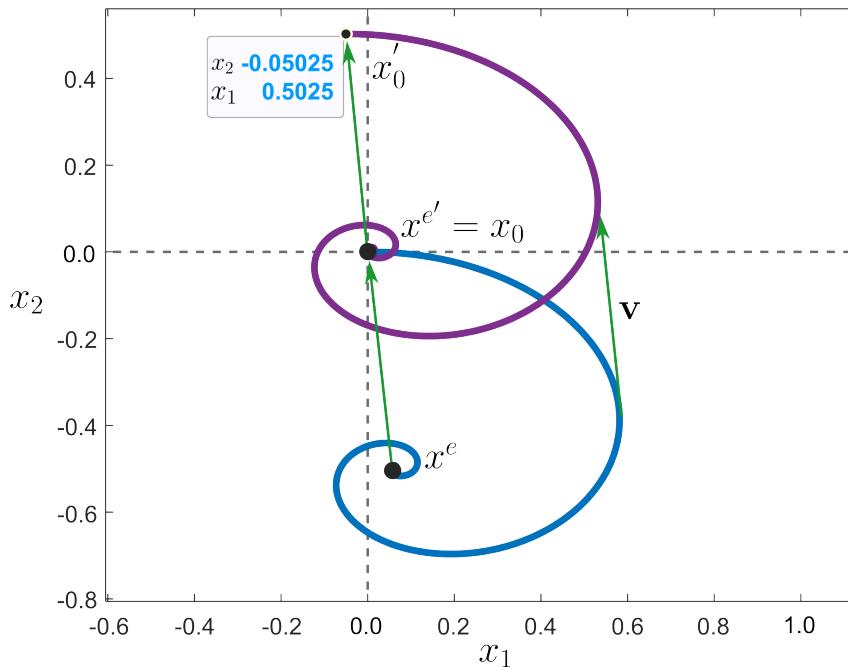
Rozwiązanie równania stanu układu niecałkowitego rzędu pozwala na wykonanie wspomnianej wcześniej translacji (3.21) o wektor:

$$\mathbf{T}(\mathbf{x}) = \mathbf{x} - \mathbf{v}, \quad (3.21)$$

co umożliwia utworzenie sztucznych warunków początkowych. Dzięki temu zabiegowi wszystkie definicje różniczkowania ułamkowego rzędu stają się zgodne, a układ uzyskuje określone warunki początkowe, co umożliwia analizę stabilności z wykorzystaniem bezpośredniej metody Lapunowa. Jak przedstawiono wcześniej, zarówno analityczna, jak i numeryczna pochodna prowadzą do tego samego wyniku, co zilustrowano na Rysunku 3.7.

Rysunek 3.8 ilustruje operację translacji w przestrzeni stanu. Jedną z istotnych wad tej metody jest wprowadzenie sztucznych warunków początkowych. Należy pamiętać, że rzeczywisty układ nie rozpoczyna ruchu od takich zmodyfikowanych warunków, co może prowadzić do róż-

nic w analizie dynamicznej. Wyniki uzyskane przy sztucznych warunkach początkowych mogą nie odzwierciedlać rzeczywistej stabilności układu w praktycznych warunkach pracy. Jednak jeśli założy się, że układ funkcjonuje w przebadanej przestrzeni sterowań i rozpoczęta pracę z zerowymi warunkami początkowymi, to jego stabilność może zostać prawidłowo określona w ramach tej ograniczonej przestrzeni. Taka analiza odzwierciedla sposób funkcjonowania układu w warunkach pracy, pod warunkiem, że operacje sterowania i ich ograniczenia zostały poprawnie uwzględnione.



Rysunek 3.8: Translacja w przestrzeni stanu

Przestrzeń jest zbiorem translacji nieskończoną liczbą punktów zerowych.

3.7. FUNKCJE LAPUNOWA UKŁADÓW UŁAMKOWYCH

Wcześniej opisano klasyczne podejście do optymalizacji, oparte na równaniu Eulera-Lagrange'a (2.9). W przypadku pochodnych rzędu ułamkowego, źródła [25] wskazują, że można rozpatrywać zaktualizowany problem, opisany równaniem:

$$\frac{d}{dt} \left(\frac{\partial L}{\partial \mathcal{D}^\alpha x} \right) - \frac{\partial L}{\partial x} - \frac{\partial f}{\partial \mathcal{D}^\alpha x} = 0 \quad (3.22)$$

Wprowadzenie rachunku różniczkowo-całkowego rzędu ułamkowego do przestrzeni stanu stanowi wyzwanie zarówno od strony teoretycznej, jak i fizycznej. W szczególności nasuwa się pytanie, czy przestrzeń fizyczna może posiadać właściwości pamięci, skoro klasyczne równanie Eulera-Lagrange'a, oparte na całkowitych pochodnych, skutecznie opisuje szeroką klasę układów fizycznych. Fundamentalnym problemem wynikającym z wprowadzenia takiego po-

dejścia jest przypisanie pamięci wymiarom przestrzeni stanu. Jeśli założymy, że przestrzeń może posiadać pamięć, implikowałoby to, że jej stan zależy od wcześniejszych konfiguracji, co stoi w sprzeczności z powszechnie przyjętymi fizycznymi właściwościami przestrzeni. Drugim ważnym aspektem jest zastosowanie funkcji Lapunowa. W klasycznym formalizmie kryterium stabilności opiera się na pochodnej całkowitej, co umożliwia jednoznaczna interpretację zachowania układu. Wprowadzenie pochodnej rzędu ułamkowego prowadzi jednak do potencjalnych trudności. Przykładowo, funkcja Lapunowa mogłaby wskazywać na malejącą wartość pochodnej, uwzględniając przeszłe stany układu, mimo że rzeczywisty układ utraciłby stabilność. Tego rodzaju zjawiska komplikują zastosowanie rachunku różniczkowo-całkowego rzędu ułamkowego w przestrzeni stanu. Chociaż tego typu podejście może upraszczać modelowanie matematyczne niektórych układów, powyższe ograniczenia podważają jego fizyczną sensowność w zastosowaniach praktycznych.

Podczas analizy definicji numerycznej Grünwalda-Letnikova zauważono ciekawą zależność dotyczącą współczynników pamięci funkcji, co zostało przedstawione w Listingu 3.4. Linie 1–4 ilustrują wartości współczynników dla układu całkowitego rzędu, natomiast linie 5–9 odnoszą się do rzędu $\alpha = 0.1$. W przypadku różnic skończonych dla układu całkowitego rzędu wartości te opisują standardowe różnice kolejnych elementów. Jednak zmniejszenie rzędu prowadzi do stopniowego zmniejszania wartości kolejnych współczynników, które zbliżają się do zera. W kontekście modeli rachunku różniczkowego i ich zastosowań w fizyce, taki efekt oznacza, że wraz ze zmniejszaniem rzędu układ zmienia swój charakter, aż do osiągnięcia postaci członu proporcjonalnego.

Listing 3.4: współczynniki pamięci

```

1 %ans =
2
3 %      1     -1      0      0      0
4 %-----
5 % >>back_v(0.1, 4)
6
7 %ans =
8
9 %      1.0000   -0.1000   -0.0450   -0.0285   -0.0207

```

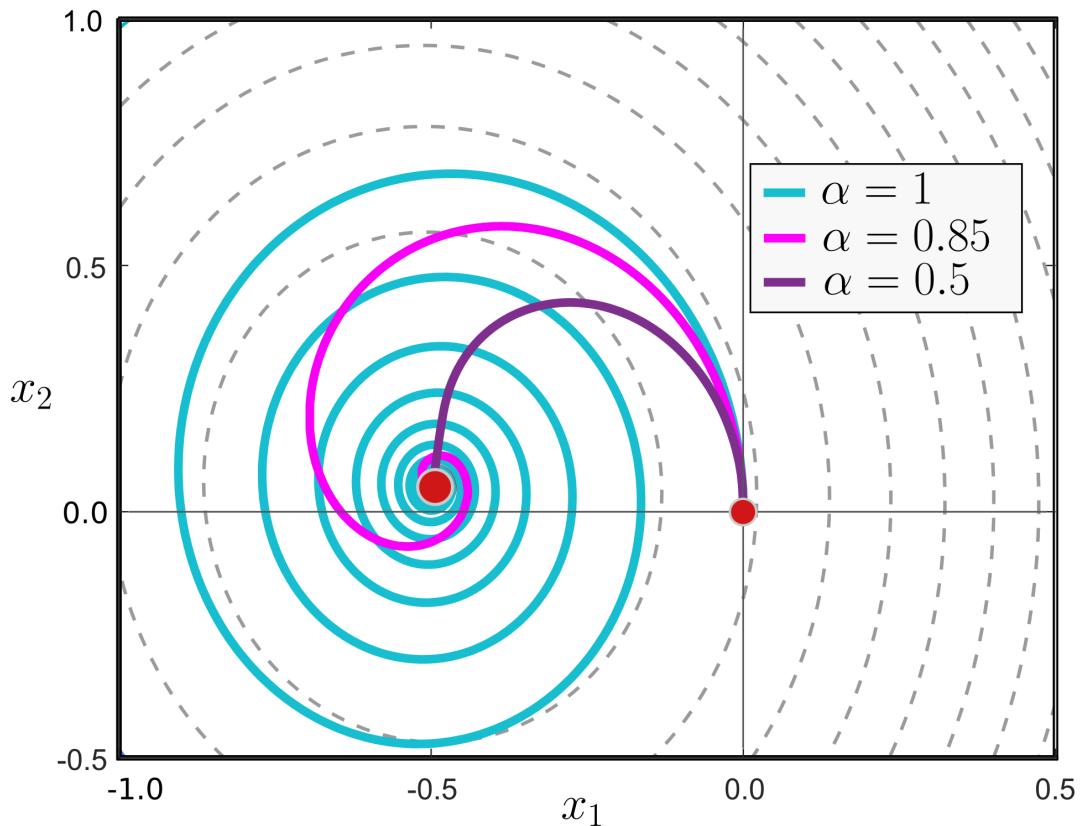
Warto również zwrócić uwagę, że idealny człon czysto proporcjonalny w rzeczywistości nie istnieje. Funkcjonowanie niektórych zjawisk, z uwagi na szybkość ich przebiegu, może wykazywać charakter proporcjonalny w odniesieniu do rozdzielczości próbkowania podczas obserwacji. Jednak każda reakcja fizyczna, zgodnie z zasadami mechaniki klasycznej, jest związana z pewnym opóźnieniem, wynikającym chociażby z ograniczonej prędkości propagacji sygnałów, np. prędkości światła. Na przykład, w przypadku zadania skokowego sygnału wejściowego na dany obiekt, uzyskanie wyjściowego sygnału w pełni odzwierciedlającego wejście wymaga-

łoby zastosowania nieskończonych wartości prędkości i przyspieszeń, co jest fizycznie niemożliwe. Tego rodzaju problemy często pojawiają się podczas modelowania nieciągłych trajektorii za pomocą funkcji sklejanych (spline). Aby je rozwiązać, należy uwzględnić warunki ciągłości prędkości i przyspieszeń, co pozwala na eliminację nieciągłości i uzyskanie bardziej realistycznych wyników.

WNIOSZEK 3.1 Na podstawie analizy współczynników pamięci można stwierdzić, że układ dynamiczny, którego rząd zmniejsza się do zera, dąży do przybrania postaci członu proporcjonalnego. W związku z tym funkcja Lyapunowa całkowitego rzędu może opisywać także dynamikę członu o rzędzie w zakresie $\alpha \in [0, 1]$.

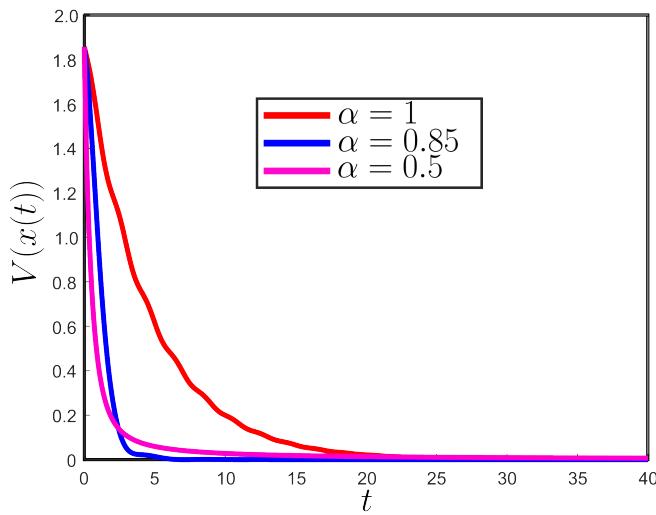
PRZYKŁAD 3.1

Układ (2.1) z macierzą stanu (2.4), którego wektor wymuszeń ma postać $B = [0 \ 1]^T$, poddano sterowaniu $u = \mathbb{1}(t)$, z zerowych warunków początkowych, dla trzech rzędów $\alpha = 1$, $\alpha = 0.85$, $\alpha = 0.5$. Położenie równowagi dynamicznej obliczono, przy użyciu funkcji `fsoolve`, zgodnie z równaniem (2.20). Translacje nie mają wpływu na działanie funkcji Lapunowa, jak pokazano w poprzednim podrozdziale, dlatego funkcja została przesunięta w położenie równowagi dynamicznej. Otrzymane rezultaty zestawia Rysunek 3.9.



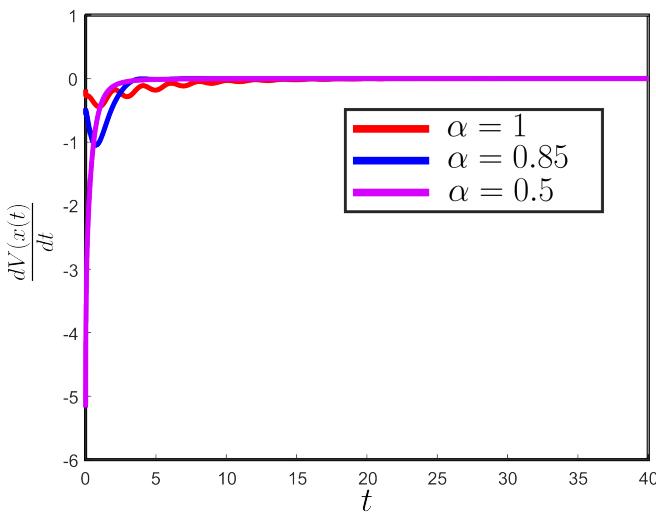
Rysunek 3.9: Wpływ zmiany rzędu modelu na dopasowanie trajektorii do poziomów funkcji Lapunowa

Przytoczony przykład potwierdza uwagę. Układy niecałkowitego rzędu, dla których rząd $\alpha \in [0, 1]$, mogą być opisane funkcją Lapunowa wyznaczoną dla odpowiadającego im układu całkowitego rzędu. Podczas działania wymuszenia, w układach stacjonarnych i liniowych, można przesunąć kwadratową funkcję Lapunowa do punktu równowagi dynamicznej. Wartość przesuniętej funkcji w punkcie zerowym przestrzeni stanu, jest modułem ε . Rysunek 3.10 przedstawia funkcje Lapunowa $V = x^T P x$, gdzie P jest dobierane wyłącznie dla układu o rzędzie całkowitym.



Rysunek 3.10: Funkcje Lapunowa układów o rzędach niecałkowitych

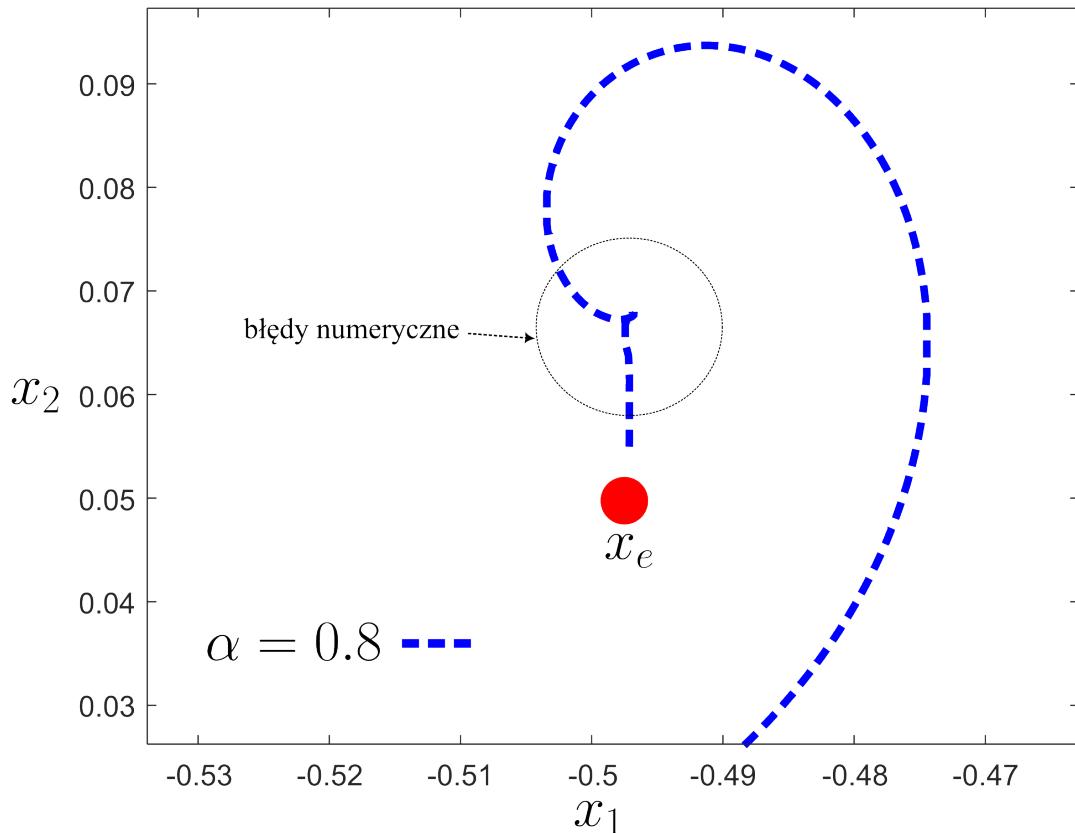
Pochodne całkowite funkcji Lapunowa dla układów o rzędzie niecałkowitym zostały przedstawione na Rysunku 3.11. Otrzymane wyniki są zbieżne do położenia równowagi. Występują jedynie błędy zapewne związane z aproksymacją, co przedstawia Rysunek 3.12.



Rysunek 3.11: Całkowite pochodne funkcji Lapunowa układów o rzędach niecałkowitych

Uwaga jest uzasadniona i funkcje Lapunowa z wyłączeniem błędów numerycznych speł-

niają założenia teorii Lapunowa, jednak nie są rozwiązaniem optymalnym, w wyniku czego algorytmy oparte na metodzie Lapunowa mogą wykazywać pewne trudności. Jednym problemem związanym z poszukiwaniem funkcji energii są błędy numeryczne rachunku, prawdopodobnie związane z akumulacją pamięci lub z błędami aproksymacji, które mają wpływ na końcową część pochodnej funkcji Lapunowa. Sprawiają, że pomimo posiadania ujemnej pochodnej przez większość czasu trwania sygnału, przez kilka próbek, pomimo dążenia do punktu równowagi, pochodna będzie w kilku miejscach dodatnia.



Rysunek 3.12: Błędy numeryczne

3.8. OPTYMALNE FUNKCJE LAPUNOWA - METODA SIMPLEKS

Parametry funkcji Lapunowa (2.18) układu autonomicznego, można odszukać minimalizując funkcjonał kosztu:

$$J = \int_0^\infty [x^T Q x] dt, \quad (3.23)$$

który można zapisać w postaci równoważnej:

$$J = x_0^T P x_0 - x_0^T P x_0 + \int_0^\infty [x^T Q x] dt, \quad (3.24)$$

a następnie część z minusem wprowadzić pod całkę:

$$J = x_0^T P x_0 + \int_0^\infty \left[\frac{d}{dt} (x^T P x) + x^T Q x \right] dt. \quad (3.25)$$

Można tak zrobić, ponieważ:

$$\int_0^\infty \frac{d}{dt} (x^T P x) dt = [x^T P x]_0^\infty = 0 - x_0^T P x_0 \quad (3.26)$$

w nieskończoności wektor stanu układu autonomicznego dąży do zera, natomiast w czasie równym zero wektor stanu jest równy warunkom początkowym. Równoważnym zapisem pochodnej $\frac{d}{dt}(x^T P x)$ jest $\dot{x}^T P x + x^T P \dot{x}$, z tego względu funkcjonał kosztu rozwija się dalej do postaci:

$$J = x_0^T P x_0 + \int_0^\infty [\dot{x}^T P x + x^T P \dot{x} + x^T Q x] dt. \quad (3.27)$$

W wypadku liniowych, stacjonarnych układów o rzędzie całkowitym pochodną \dot{x} jest równanie stanu, co upraszcza rozwiązań. Jednak w wypadku układów o rzędzie niecałkowitym problem jest bardziej rozbudowany. Rozwiązań równania stanu autonomicznego układu o rzędzie niecałkowitym jest szereg Mittaga-Lefflera [37]:

$$x(t) = E_\alpha(At) \cdot x_0 = \sum_{k=0}^{\infty} \frac{(A)^k t^{k\alpha}}{\Gamma(k\alpha + 1)} x_0, \quad (3.28)$$

którego pochodną po czasie, dla każdego składnika jest:

$$\dot{x}(t) = \sum_{k=0}^{\infty} \frac{k\alpha A^k t^{k\alpha-1}}{\Gamma(k\alpha + 1)} x_0, \quad (3.29)$$

co zdecydowanie komplikuje dalsze przekształcenia związane z funkcjonałem kosztu. Z tego względu rozsądnie jest odszukać minimalne rozwiązanie funkcjonału kosztu metodą numeryczną np. simpleks, która jest efektywna i skuteczna. Ostatecznie funkcjonał kosztu wymaga jeszcze jednej zmiany. Trzeba zauważyć, że w procesie optymalizacji pod wpływem obliczeń wartość funkcji celu, pochodna $\frac{d}{dt}(x^T P x)$ może być ujemna. Z tego względu należy zastosować moduł pod całkowaniem, żeby funkcjonał posiadał minimum, możliwe do określenia przez algorytm simpleks:

$$J^* = x_0^T P x_0 + \int_0^\infty |[\dot{x}^T P x + x^T P \dot{x} + x^T Q x]| dt. \quad (3.30)$$

w innym wypadku optymalizacja nie przebiegnie pomyślnie, ponieważ funkcja nie będzie miała określonego minimum. Ostatnim etapem jest dyskretyzacja funkcji celu, żeby było możliwe

wprowadzenie jej do programów obliczeniowych:

$$J^* = x_0^T Px_0 + \sum_{i=0}^N \left| \left[\frac{\Delta x(i)}{\Delta T}^T Px(i) + x(i)^T P \frac{\Delta x(i)}{\Delta T} + x(i)^T Q x(i) \right] \Delta T \right|, \quad (3.31)$$

gdzie:

- N – liczba próbek sygnałów,
- i – bieżąca próbka,
- ΔT – okres próbkowania,
- $\frac{\Delta x(i)}{\Delta T} = \frac{x(i)-x(i-1)}{\Delta T}$

przy czym należy pamiętać, że wyprowadzenie jest formą w przybliżeniu równoważną:

$$J = \int_0^\infty x^T Q x = x_0^T Px_0 + \int_0^\infty [\dot{x}^T Px + x^T P \dot{x} + x^T Q x] dt. \quad (3.32)$$

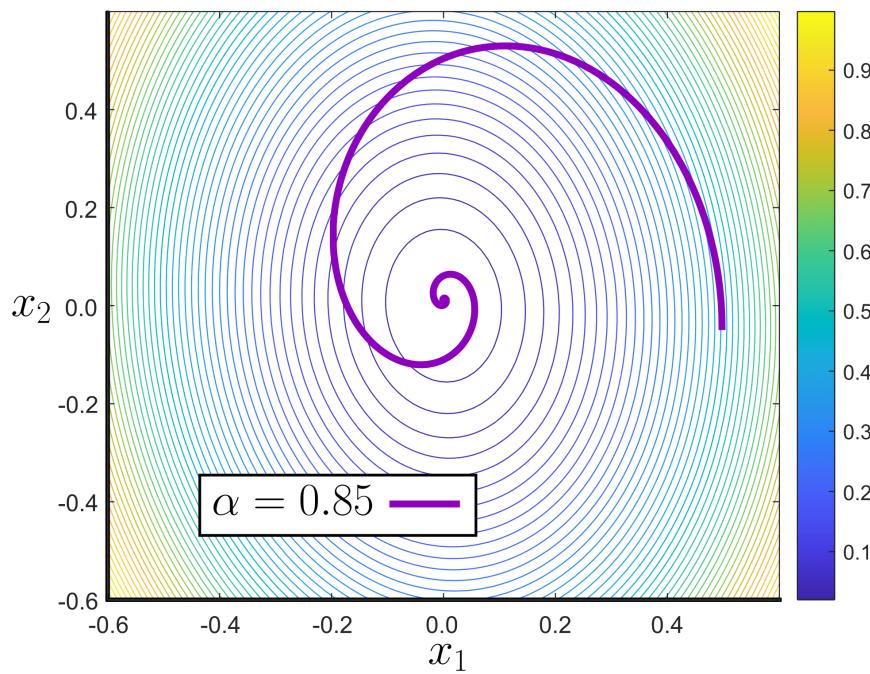
Zadanie można rozwiązać przy użyciu metody Nealder-Mead (simpleks) [39, 40]. Rozwiążanie otrzymane w metodzie, z założoną dokładnością, odnajduje minimum jednostkowego funkcjonału kosztu (3.31). Metoda dopasowuje funkcję $x^T Px$, w taki sposób, że określa minimalny koszt przejścia naturalnej trajektorii układu przez przestrzeń stanu, co jest zgodne z zasadą minimalnego działania.

PRZYKŁAD 3.2

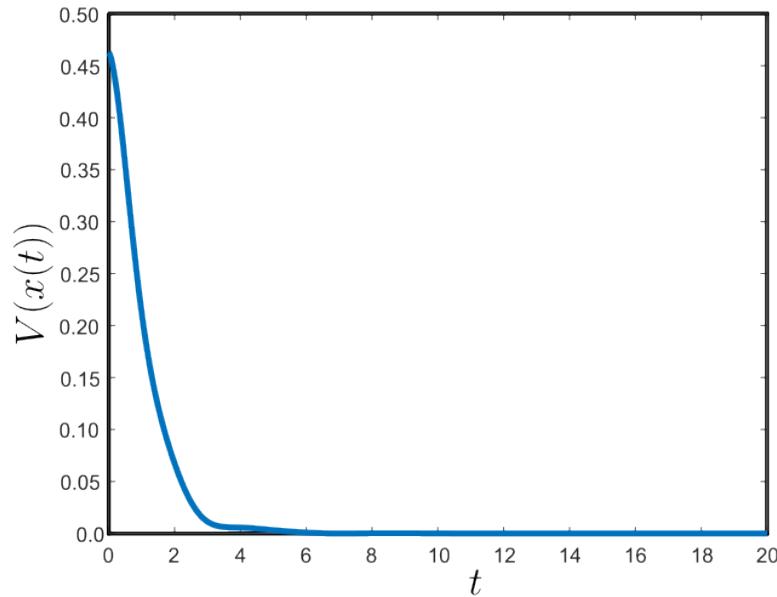
Układ (2.1) z macierzą stanu (2.4), którego wektor wymuszeń ma postać $B = [0 \ 1]^T$, poddano sterowaniu $u = \mathbb{1}(t)$, z zerowych warunków początkowych, dla dwóch rzędów $\alpha = 0.85$, $\alpha = 0.5$. Położenie równowagi dynamicznej obliczono, przy użyciu funkcji `fsoolve`, zgodnie z równaniem (2.20). Trajektorię układu przesunięto, przy użyciu wcześniej opisanej translacji, tak że punkt równowagi dynamicznej znajduje się w początku układu współrzędnych. Do poszukiwań macierzy P wykorzystano funkcję `fminsearch`, która minimalizuje funkcjonał kosztu (3.31). Rezultaty optymalizacji dla układu o rzędzie $\alpha = 0.85$ ilustruje Rysunek 3.13. Można na nim zauważyć, że optymalizacja odwzorowuje nawet obrót funkcji, zachowując odpowiednie dopasowane przestrzeni, nie tylko przez rozciągnięcie w osiach. Parametry funkcji Lapunowa, dla rzędu $\alpha = 0.85$ wynoszą:

$$P|_{(\alpha=0.85)} = \begin{bmatrix} 1.874 & 0.222 \\ -0.097 & 0.826 \end{bmatrix}. \quad (3.33)$$

Rysunek 3.14 przedstawia wartości funkcji otrzymane podczas optymalizacji, dla rzędu $\alpha = 0.85$. Warto zauważyć, że otrzymane wartości nie są dodatnie w każdej chwili, co ma związek z niedokładnością odwzorowania trajektorii przedstawioną na Rysunku 3.12.

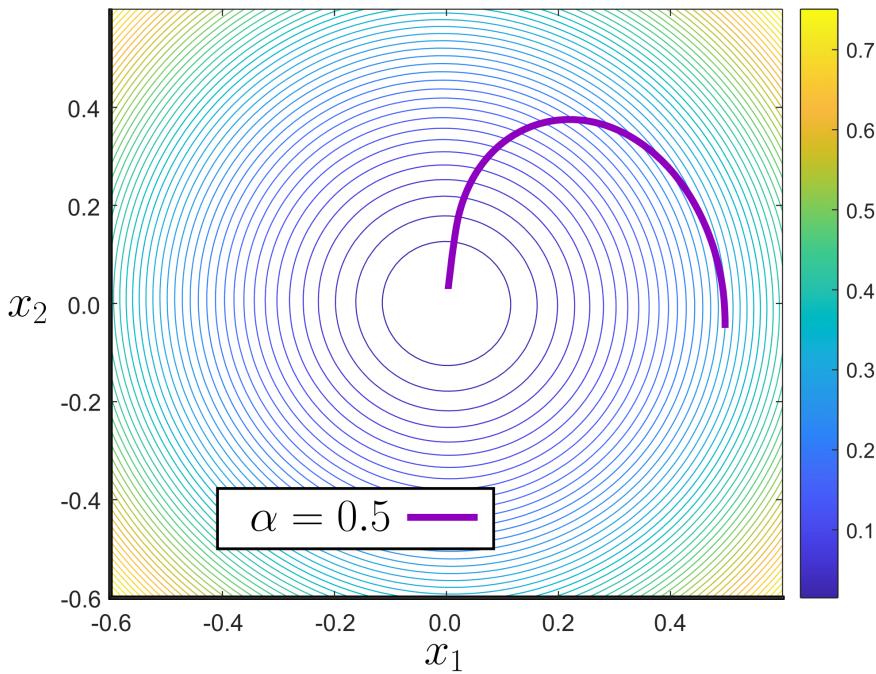


Rysunek 3.13: Trajektoria układu niecałkowitego rzędu $\alpha = 0.85$ i poziomice optymalnej funkcji Lapunowa.



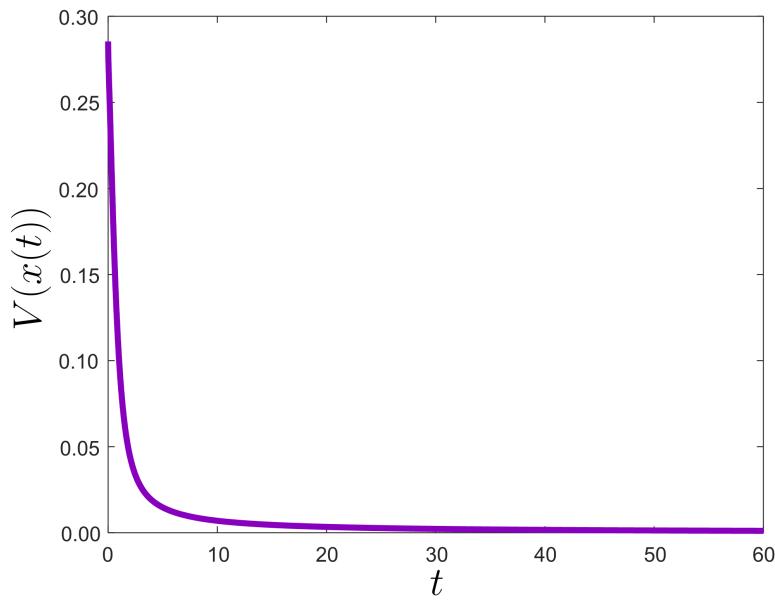
Rysunek 3.14: Optymalna funkcja Lapunowa układu o rzędzie $\alpha = 0.85$

Optymalizacja funkcjonuje właściwie także dla rzędu $\alpha = 0.5$, co przedstawia Rysunek 3.15.



Rysunek 3.15: Trajektoria układu niecałkowitego rzędu $\alpha = 0.5$ i poziomice optymalnej funkcji Lapunowa.

Rysunek 3.16 ilustruje wartości optymalnej funkcji Lapunowa. W przedstawionym wypadku nie występują błędy numeryczne. Układ dąży do stanu równowagi dynamicznej, w wyniku czego wartości optymalnej funkcji pozostają dodatnie, spełniając warunek (2.24).



Rysunek 3.16: Optymalna funkcja Lapunowa układu o rzędzie $\alpha = 0.5$

Parametry funkcji Lapunowa, dla rzędu $\alpha = 0.5$ wynoszą:

$$P|_{(\alpha=0.5)} = \begin{bmatrix} 1.147 & 0.015 \\ 0.021 & 0.941 \end{bmatrix}. \quad (3.34)$$

3.9. OPTYMALIZACJA WARUNKOWA FUNKCJI LAPUNOWA

W poprzednim podrozdziale została przedstawiona metoda doboru funkcji Lapunowa układów ułamkowych rzędów, przy wykorzystaniu funkcjonału kosztu powszechnie wykorzystywanego w problemie optymalnego sterowania, który zapewnia optymalne dopasowanie funkcji do dynamiki modelu. Metoda wymaga jednak zaangażowania większych zasobów obliczeniowych, ze względu na wykorzystanie metody Nealdera-Meada, co może się nie sprawdzić w wypadku, jeśli funkcja Lapunowa byłaby obliczana dynamicznie na potrzeby algorytmu sterowania np. przy wykorzystaniu mikrokontrolera. W bieżącym podrozdziale zostanie zaproponowana alternatywna metoda dopasowania funkcji Lapunowa do dynamiki układów niecałkowitego rzędu, która wymaga użycia mniejszych zasobów obliczeniowych.

Macierz parametrów funkcji P , optymalną dla dynamiki układu całkowitego rzędu, wyznaczoną metodą (2.15), można dostosować tak, aby minimalnie spełniała warunek (2.24). W tym celu należy wykorzystać iloczyn Hadamarda:

$$P(n) = P(n-1) \odot D = P \odot \begin{bmatrix} d(n) & 1 \\ d(n) & 1 \end{bmatrix}, \quad (3.35)$$

$$P(n) = P(n-1) \odot D = P \odot \begin{bmatrix} d(n) & d(n) \\ 1 & 1 \end{bmatrix}, \quad (3.36)$$

gdzie \odot oznacza iloczyn Hadamarda na parametrach nowej funkcji Lapunowa, a $d(n)$ jest współczynnikiem optymalizacji:

$$d(n) = d(0) + n\kappa, \quad (3.37)$$

gdzie $d(0)$ jest wartością początkową, a κ odpowiada długości kroku optymalizacji. W związku z tym macierz $P(0)$ odpowiada wartościom macierzy P otrzymanej dla rzędu całkowitego. Istotne jest, aby początkowa wartość współczynnika $d(0)$ była dokładnie równa jeden. Zapewnia to zachowanie kluczowych właściwości charakterystycznych dla układów o rzędach całkowitych. Dla układów liniowych o rzędzie $\alpha < 1$ długość kroku optymalizacji jest ujemną liczbą rzeczywistą $\kappa \in \mathcal{R}^- \setminus \{0\}$. W sytuacjach, gdy n dąży do nieskończoności, warunek zakończenia określa pierwszy zestaw parametrów, przed którym (2.24) nie jest spełniony. Metoda zmienia "rozciągnięcie" funkcji w przestrzeni stanu. Proponowana struktura iloczynu wynika z dalszych rozważań.

Kolejnym aspektem do zbadania jest to, czy proces optymalizacji zachowuje dodatnią półokreśloność funkcji Lapunowa i opracowanie przynajmniej częściowego dowodu działania

metody. Forma kwadratowa jest dodatnio półokreślona, gdy wszystkie główne minory wiodące są nieujemne. Zaproponujmy macierz P :

$$P = \begin{bmatrix} p_1 & p_2 \\ p_3 & p_4 \end{bmatrix}. \quad (3.38)$$

Kryterium konieczne zapewniające, że kwadratowa funkcja Lapunowa pozostaje dodatnio półokreślona, jest następujące:

$$p_1 \geq 0 \quad \wedge \quad p_1 p_4 \geq p_2 p_3. \quad (3.39)$$

Ponadto, jeśli zachodzi warunek $d(n) > 0$, to warunki te przekształcają się w:

$$p_1 d(n) \geq 0 \quad \wedge \quad p_1 p_4 d(n) \geq p_2 p_3 d(n). \quad (3.40)$$

WNIOSK 3.2

Zakładając, że w procesie optymalizacji κ jest wartością kroku mniejszą od zera i n rośnie. Wtedy $d(n)$ może być wartością mniejszą od zera, co nie spełnia warunku (3.40). Wtedy funkcja $V(x(t))$ może przyjmować wartości ujemne, a jej pochodna dodatnie, co nie spełnia warunku (2.24). Jeśli algorytm zakończył się na granicznej wartości parametrów i jeżeli pierwotna forma kwadratowa jest dodatnio półokreślona, to każda funkcja wyprowadzona z tej formy poprzez optymalizację również zachowa tę właściwość i zostanie zachowany warunek (2.24).

PRZYKŁAD 3.3 Rozważmy liniowy układ o rzędzie ułamkowym $\alpha = 0.8$, modelowany aproksymacją Oustaloupa z następującymi parametrami, podlegający sterowaniu $u(t) = \mathbb{1}(t)$ oraz $x_0 = [0 \ 0]^T$ i $B = [0 \ 1]^T$:

$$A = \begin{bmatrix} -0.1 & -1 \\ 2 & -0.1 \end{bmatrix}.$$

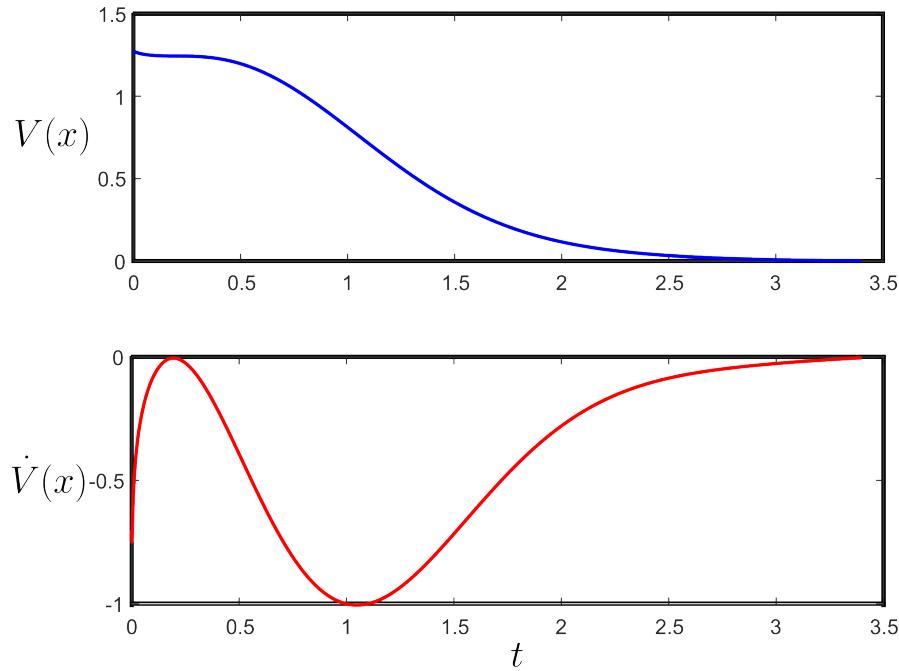
Z równania (2.15) otrzymujemy:

$$P = \begin{bmatrix} 7.487 & 0.124 \\ 0.124 & 3.756 \end{bmatrix}.$$

Następnie optymalizacja (3.35) prowadzi do:

$$P' = \begin{bmatrix} 5.128 & 0.124 \\ 0.085 & 3.756 \end{bmatrix},$$

gdzie P' jest macierzą uzyskaną z algorytmu (3.35). Rysunek 3.17 ilustruje funkcję otrzymaną przy wykorzystaniu proponowanej optymalizacji warunkowej. Zaletą proponowanej metody jest wymóg mniejszych zasobów obliczeniowych w porównaniu do algorytmu simpleks, natomiast jej wadą jest to, że otrzymana funkcja nie jest optymalna względem funkcjonału kosztu (3.31).



Rysunek 3.17: Przykładowa funkcja Lapunowa otrzymana w wyniku optymalizacji warunkowej

3.10. FUNKCJE LAPUNOWA NA PRZEDZIALE CZASU

Bieżący podrozdział zwraca uwagę na problemy związane z numeryczną implementacją szeregu Mittaga-Lefflera (ML), który jest rozwiązaniem autonomicznego układu dynamicznego niecałkowitego rzędu:

$$x(t) = \sum_{k=0}^{\infty} \frac{A^k t^{k\alpha}}{\Gamma(k\alpha + 1)} x_0, \quad (3.41)$$

jednocześnie próbując rozwiązać problem optymalizacji związany z uogólnieniem algebraicznego równania Riccatiego na układy dynamiczne niecałkowitego rzędu. Analityczne wyprowadzenie sprawnego rozwiązania zadania optymalizacji (3.23) potencjalnie doprowadzi do rozwiązań zwracających parametry funkcji Lapunowa układów liniowych, w sposób który mniej angażuje zasoby komputerowe. W obecnie zaprezentowanej postaci szereg ML można zaimplementować w dowolnym języku programowania, który pozwala na wykonywanie obliczeń. W ramach podrozdziału szereg został zaimplementowany przy wykorzystaniu języka C++ [16], ponieważ nie musi być interpretowany do C tak jak dzieje się w wypadku Pythona, przez co jest to rozwiązanie, które może wykonywać obliczenia szybciej. Biblioteka Eigen [17] umożliwia wykonywanie działań arytmetycznych, które upraszczają implementację rozwiązania do układu zawierającego dowolną kwadratową macierz stanu A . Otrzymane rozwiązania są ewaluowane i porównywane do rozwiązań otrzymywanych z biblioteki FOTF pakietu Matlab, ponieważ jako jedyna umożliwia zadanie warunków początkowych. Model porównawczy zo-

stał już przedstawiony w obecnym rozdziale 3.5. Listing 3.5 przedstawia program, który oblicza wartości szeregu ML w czasie, przy wykorzystaniu języka C++.

Listing 3.5: Obliczenia szeregu Mittaga-Lefflera w C++

```
1 #include "mex.h"
2 #include "Eigen/Core"
3 #include <cmath>
4
5 using namespace std;
6 using namespace Eigen;
7
8 MatrixXd matrix_pow(const MatrixXd &m, int n) {
9     MatrixXd result = MatrixXd::Identity(m.rows(), m.cols());
10    for (int i = 0; i < n; ++i) {
11        result *= m;
12    }
13    return result;
14 }
15
16 MatrixXd mittag_leffler_matrix(const MatrixXd &A, double alpha,
17                                 double t, int max_terms) {
18     int rows = A.rows();
19     int cols = A.cols();
20
21     MatrixXd result = MatrixXd::Zero(rows, cols);
22
23     for (int k = 0; k < max_terms; ++k) {
24         double denominator = 1.0 / tgamma(alpha * k + 1);
25         MatrixXd power_of_A = matrix_pow(A * pow(t, alpha), k);
26         result += power_of_A * denominator;
27     }
28
29     return result;
30 }
```

Narzędzie `mex.h` jest związane z pakietem Matlab i umożliwia zbudowanie funkcji C++ w postaci funkcji Matlab, dzięki czemu można w nim zastosować funkcję napisaną w C++.

Listing 3.6: Implementacja obliczeń szeregu w pakiecie Matlab

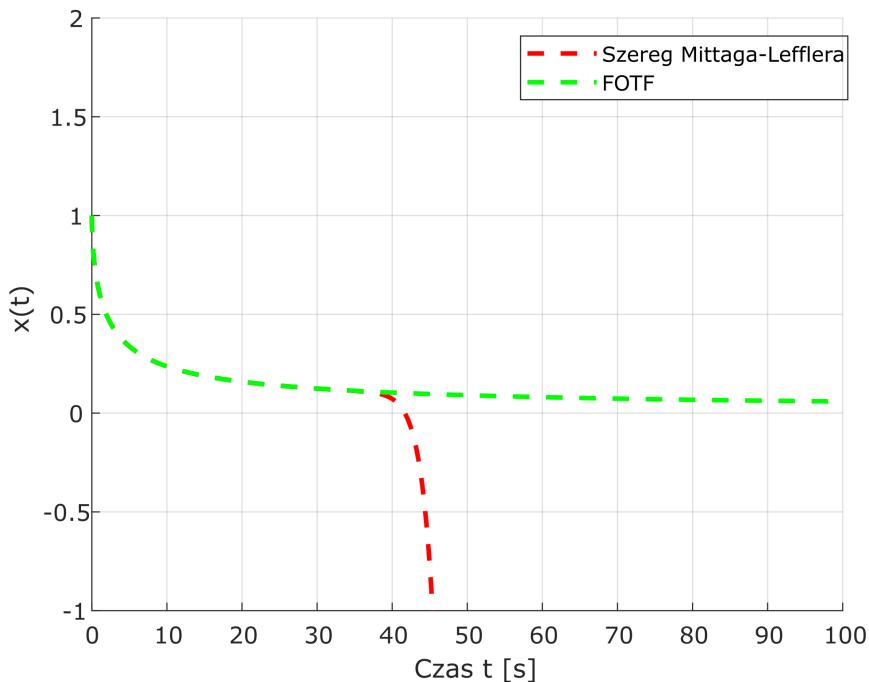
```
1 alpha = 0.6;
2 lambda = -0.5;
3 dt = 0.01;
4 kmax = 300;
```

```

5 x0 = 1;
6 T = 100;
7
8 t = 0:dt:T-dt;
9
10 for i=0:1:length(t)-1
11     x_true(i+1) = mittag_leffler(lambda, alpha, t(i+1), kmax);
12 end

```

Rysunek 3.18 przedstawia problem związany z implementacją szeregu, którego wyrazy są skalowane z czasem. Szereg rozbiega się z czasem, dlatego że zmienne stosowane w informatyce mogą zawierać jedynie skończone wartości, natomiast podnoszenie licznika do skończonej potęgi tworzy dużą liczbę, która dodatkowo jest skalowana przez czas. Nawet jeśli z początku obliczenia prowadzone są z wystarczającą dokładnością, ponieważ pokrywają się z biblioteką FOTF, w większym czasie osiągane są wartości znajdujące się na końcu zakresu zmiennej, w wyniku czego utracona zostaje precyzja obliczeń. Mianownik również szybko narasta, przez co komputer w ostateczności przypisuje mianownikowi wartość zmiennej podwójnej precyzji inf , natomiast licznik w przypadku układu stabilnego dąży do zera. Wynikiem takiego prowadzenia obliczeń jest zbliżanie się do wykonania przez procesor operacji $\frac{0}{\text{inf}}$. Szereg rozbiega się numerycznie, aż w ostateczności zaczyna przyjmować wartości NAN - not a number.



Rysunek 3.18: Otrzymane rozwiązanie szeregu ML wraz z porównaniem z biblioteki FOTF

F. Mainardi [51] przedstawia właściwości szeregu, między innymi potencjalne rozwiązanie problemu związanego z uzyskaniem rozwiązania szeregu. Jest nim asymptotyczna aproksy-

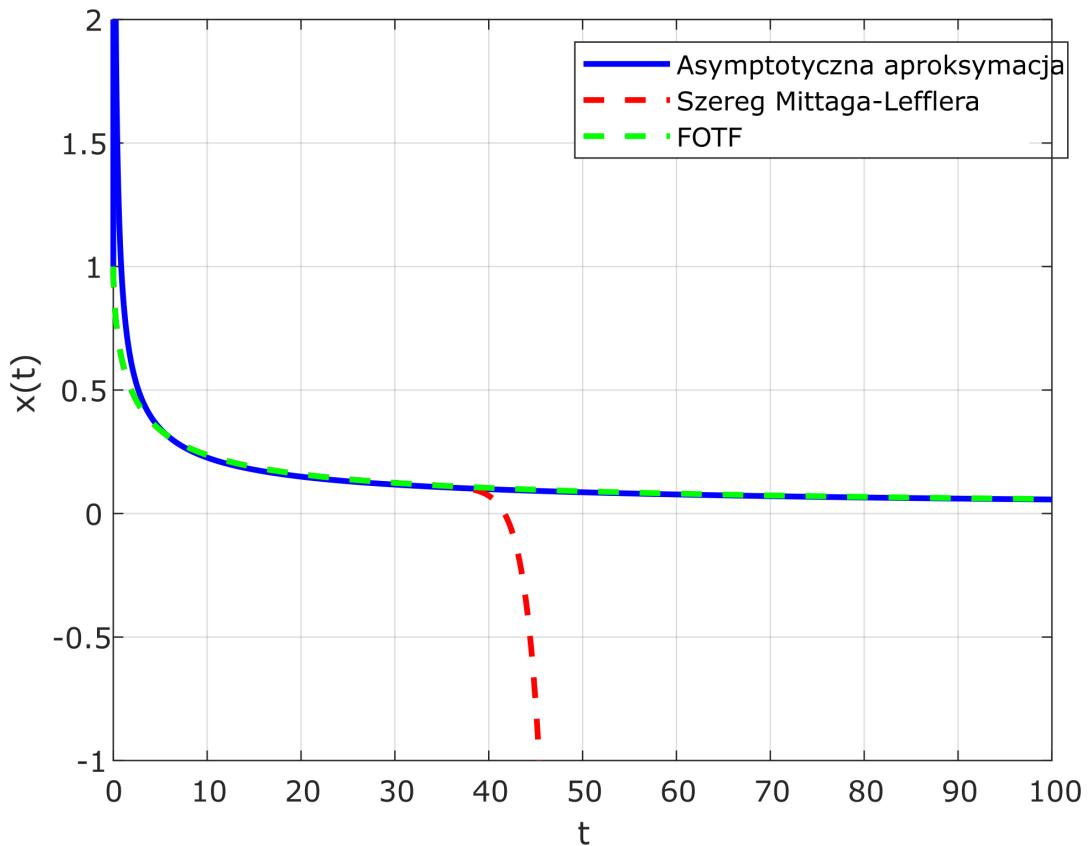
macja:

$$E_\alpha(\lambda t^\alpha) = \sum_{k=0}^{\infty} \frac{(\lambda t^\alpha)^k}{\Gamma(k\alpha + 1)} \approx -\frac{1}{\lambda t^\alpha \Gamma(1-\alpha)}. \quad (3.42)$$

gdzie:

- $\alpha \in (0, 1)$

Przykładowe rozwiązanie zestawione na Rysunku 3.19 jest wykonywane dla szeregu, w którym λ jest jedną wartością, czyli w wypadku odtwarzania rozwiązania równania stanu układu pierwszego rzędu o niecałkowitej pochodnej $\alpha = 0.6$. W związku z występowaniem asymptoty pojawiają się jednak dodatkowe problemy podczas symulacji odpowiedzi układu, ponieważ aproksymacja dla małych czasów, bliskich zeru asymptotycznie dąży do nieskończoności, pomimo zastosowania warunku początkowego. Można jednak zastosować podejście mieszane, które polega na tym, że dla małych czasów obliczane są wartości obliczone przez zastosowanie skończonej sumy szeregu, natomiast dla większych czasów można zastosować asymptotyczną aproksymację. Takie rozwiązanie rodzi jednak dodatkowy problem związany z czasem, w którym ma zachodzić przejście między obliczaniem szeregu w obliczeniu aproksymacji.



Rysunek 3.19: Porównanie aproksymacji z biblioteką FOTF i numerycznymi obliczeniami szeregu

Analiza struktury aproksymacji dla rzędów pochodnej układu $\alpha \in (0, 1)$ wnosi ciekawe

spostrzeżenia, ponieważ rozwiązywanie układu o macierzy stanu A pierwszego rzędu można skonstruować w następujący sposób:

$$x(t) \approx -\frac{1}{\lambda t^\alpha \Gamma(1-\alpha)} x_0. \quad (3.43)$$

Jeżeli λ jest mniejsza od zera, wtedy:

$$\lim_{t \rightarrow \infty} \left(-\frac{1}{\lambda t^\alpha \Gamma(1-\alpha)} x_0 \right) = 0, \quad (3.44)$$

niezależnie od rzędu pochodnej układu, znajdującego się w zakresie $\alpha \in (0, 1)$. Pochodną rozwiązania po czasie jest:

$$\dot{x}(t) \approx \frac{\alpha}{\lambda t^{\alpha+1} \Gamma(1-\alpha)} x_0. \quad (3.45)$$

Na ten moment można podstawić rozwiązania do funkcjonału kosztu (3.23), jednocześnie zastępując parametr λ macierzą stanu A . Funkcję podcałkową można rozpisać w następujący sposób:

$$\begin{aligned} x_0^T (A^{-1})^T & \left[\frac{\alpha}{\Gamma(1-\alpha)} t^{-\alpha-1} P t^{-\alpha} \left(-\frac{1}{\Gamma(1-\alpha)} \right) \right. \\ & + \left(-\frac{1}{\Gamma(1-\alpha)} \right) t^{-\alpha} P t^{-\alpha-1} \left(\frac{\alpha}{\Gamma(1-\alpha)} \right) \\ & \left. + \left(-\frac{1}{\Gamma(1-\alpha)} \right) t^{-\alpha} Q \left(-\frac{1}{\Gamma(1-\alpha)} \right) t^{-\alpha} \right] A^{-1} x_0. \end{aligned} \quad (3.46)$$

Części skalarne można pogrupować i uporządkować:

$$\begin{aligned} x_0^T (A^{-1})^T & \left[-\frac{\alpha}{\Gamma(1-\alpha)^2} t^{-2\alpha-1} P \right. \\ & - \frac{\alpha}{\Gamma(1-\alpha)^2} t^{-2\alpha-1} P \\ & \left. + \left(\frac{1}{\Gamma(1-\alpha)^2} \right) t^{-2\alpha} Q \right] A^{-1} x_0 \end{aligned} \quad (3.47)$$

Następnie trzeba zgrupować wyrazy podobne:

$$x_0^T (A^{-1})^T \left[-\frac{2\alpha}{\Gamma(1-\alpha)^2} t^{-2\alpha-1} P + \left(\frac{1}{\Gamma(1-\alpha)^2} \right) t^{-2\alpha} Q \right] A^{-1} x_0.$$

Części niezależne od czasu można wyciągnąć poza całkę, z tego względu cały funkcjonał przy-

biera postać:

$$J \approx x_0^T P x_0 + \frac{1}{\Gamma(1-\alpha)^2} x_0^T (A^{-1})^T \int_{t_0}^{t_1} \left[-2\alpha t^{-2\alpha-1} P + t^{-2\alpha} Q \right] dt A^{-1} x_0. \quad (3.48)$$

Na tym etapie można wykonać operację całkowania:

$$J \approx x_0^T P x_0 + \frac{1}{\Gamma(1-\alpha)^2} x_0^T (A^{-1})^T \left[t^{-2\alpha} P + \frac{t^{1-2\alpha}}{1-2\alpha} Q \right]_{t_0}^{t_1} A^{-1} x_0. \quad (3.49)$$

W dalszym ciągu należy rozpisać granice całkowania:

$$J \approx x_0^T P x_0 + \frac{1}{\Gamma(1-\alpha)^2} x_0^T (A^{-1})^T \left[(t_1^{-2\alpha} - t_0^{-2\alpha}) P + \left(\frac{t_1^{1-2\alpha}}{1-2\alpha} - \frac{t_0^{1-2\alpha}}{1-2\alpha} \right) Q \right] A^{-1} x_0. \quad (3.50)$$

Obecna postać wymusza zastosowanie założenia, że $\alpha \neq \frac{1}{2}$. Funkcja kosztu w okolicach rzędu pochodnej układu $\alpha = \frac{1}{2}$ będzie posiadała asymptotę, z tego względu w rozwiązaniach należy unikać rozwiązań układów bliskich $\alpha = \frac{1}{2}$. W oparciu o strukturę funkcji celu można również wnioskować, że rozwiązanie jest oparte na przedziale czasu. W klasycznym dowodzie, z którego wynika algebraiczne równanie Riccatiego minimalizacja funkcji celu sprowadza się do doprowadzenia funkcji do wartości jak najbliższej零. Z tego względu najlepiej jest porównać ją do zera. Przy założeniu, że $t_0 = 0$ rachunek dodatkowo się upraszcza:

$$0 = P + \frac{1}{\Gamma(1-\alpha)^2} (A^{-1})^T \left[t_1^{-2\alpha} P + \frac{t_1^{1-2\alpha}}{1-2\alpha} Q \right] A^{-1}. \quad (3.51)$$

Rozpisując i grupując wyrazy na dwie strony równania otrzymuje się:

$$\frac{t_1^{1-2\alpha}}{2\alpha-1} Q = A^T P A + \frac{t_1^{-2\alpha}}{\Gamma(1-\alpha)} A^T (A^{-1})^T P (A^{-1}) A. \quad (3.52)$$

Następnie iloczyny macierzy i jej odwrotności są macierzami jednostkowymi:

$$\frac{t_1^{1-2\alpha}}{2\alpha-1} Q = A^T P A + \frac{t_1^{-2\alpha}}{\Gamma(1-\alpha)} P. \quad (3.53)$$

Obliczenia macierzy parametrów funkcji Lapunowa można wykonać przez wykonanie wektoryzacji, na początku warto jednak zapisać parametry przy pomocy symboli:

$$A^T P A + \gamma P = R, \quad (3.54)$$

gdzie:

- $R = \frac{t_1^{1-2\alpha}}{2\alpha-1} Q,$

$$\bullet \quad \gamma = \frac{t_1^{-2\alpha}}{\Gamma(1-\alpha)}.$$

Równanie po wektoryzacji:

$$(A^T \otimes A^T + \gamma I) \text{vec}(P) = \text{vec}(R) \quad (3.55)$$

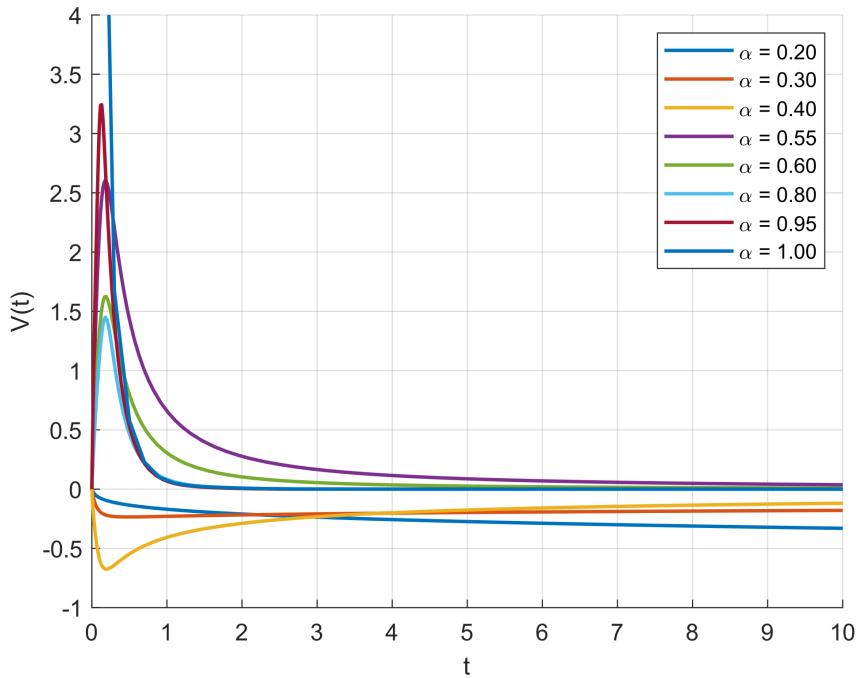
Ostateczne rozwiązanie:

$$\text{vec}(P) = (A^T \otimes A^T + \gamma I)^{-1} \text{vec}(R) \quad (3.56)$$

Można zauważyć, że w każdym czasie t zaczynając od $t_0 = 0$ funkcja Lapunowa będzie zależna od czasu (przedziału czasu):

$$V(x) = x^T P(t)x \quad (3.57)$$

Rysunek 3.20 przedstawia funkcje Lapunowa oparte na przedziale czasu, otrzymane zgodnie z równaniem (3.56), dla różnych rzędów układu. Można zauważyć, że dla małych czasów model



Rysunek 3.20: Analityczne funkcje Lapunowa

nie jest odpowiedni, natomiast sytuacja poprawia się wraz ze wzrostem przedziału czasu. W pobliżu asymptoty rozwiązania dla $\alpha \approx \frac{1}{2}$ funkcje tracą dokładność, natomiast poniżej asymptoty otrzymane rezultaty nie są odpowiednie, ponieważ wraz z malejącym rzędem α zaczynają się rozbiegać. Rzędów $\alpha > 1$ nie warto jest rozpatrywać, ponieważ zgodnie z przyjętą asymptotyczną aproksymacją argumenty funkcji Gamma wchodzą w jej ujemną część i nie są dokładne. Uzyskane funkcje Lapunowa mogą stwierdzać stabilność jedynie w wypadku modeli, w których można zadać warunki początkowe oraz rząd układu znajduje się w przedziale $\alpha \in (0.5, 1)$,

przedział jest obustronnie otwarty, ponieważ w wypadku całkowitego rzędu lepiej jest obliczyć stacjonarną funkcję Lapunowa z algebraicznego równania Riccatiego. Ostatecznie otrzymanych funkcji nie można nazwać w pełni funkcjami Lapunowa, ponieważ nie spełniają kryteriów, tzn. nie są zawsze malejące poza rzędem $\alpha = 1$.

3.11. PODSUMOWANIE ROZDZIAŁU

Rozdział omawia rachunek niecałkowitego rzędu w kontekście teorii sterowania, wprowadza wiele istotnych zagadnień teoretycznych, niezbędnych do zrozumienia dynamiki układów o tym charakterze. Jednym z głównych punktów jest problem przybliżania funkcji ciągłych za pomocą dyskretnych odpowiedników. W klasycznym rachunku całkowitego rzędu proces ten jest stosunkowo prosty i intuicyjny, ponieważ dyskretne modele można łatwo uzyskać w procesie próbkowania funkcji ciągłych, co jest szeroko opisywane w literaturze. Jednak w przypadku rachunku ułamkowych rzędów sytuacja staje się bardziej złożona.

Zagadnienie komplementarności definicji układów o dynamice niecałkowitego rzędu to kolejny aspekt poruszony w rozdziale. Układy takie wykazują specyficzne właściwości, które są różne od układów o całkowitych rzędach. W kontekście teorii sterowania istnieje potrzeba dostosowania warunków początkowych, ponieważ tradycyjne podejście do tego zagadnienia, oparte na klasycznych równaniach różniczkowych, nie jest w pełni kompatybilne z układami o dynamice ułamkowej. W przypadku układów o dynamice niecałkowitego rzędu konieczne jest wprowadzenie nowych zasad i założeń. Okazuje się, że warunki początkowe w układach fizycznych ze względu na rozbieżności modeli niecałkowitych rzędów nie mają zastosowania.

Zaprezentowanie trzech metod syntezy funkcji Lapunowa stanowi ważny wkład w rozwój narzędzi wykorzystywanych w analizie układów niecałkowitego rzędu. Każda z zaproponowanych metod posiada swoje specyficzne cechy, które mogą być zaletą w określonych sytuacjach, ale także posiadają ograniczenia, które mogą zredukować ich użyteczność w innych przypadkach. Wybór odpowiedniej metody zależy od konkretnego zastosowania, charakterystyki układu oraz wymaganej precyzji obliczeń.

Analityczne wyprowadzenie minimum problemu optymalizacji wskazuje, że funkcje Lapunowa układów o niecałkowitym rzędzie są niestacjonarne, co znaczy że parametry funkcji zmieniają się wraz z czasem. Oparcie parametrów funkcji na przedziale czasu jest zrozumiałym efektem, który jest związany z pamięcią rachunku, jednak rozwiązanie posiada wady w postaci przyjętego sposobu aproksymacji. Rozwiązanie posiada asymptoty, które dodatkowo zmniejszają dokładność metody oraz ograniczają jej stosowalność wyłącznie do rzędów z przedziału $\alpha \in (0.5, 1)$. Formalnie również otrzymane wyniki nie spełniają założeń bezpośredniej metody Lapunowa, ponieważ wartości funkcji nie zawsze maleją. Metoda wymaga poprawy i dalszych badań, których zakres zdecydowanie wykracza poza tematykę tej pracy.

4. STABILIZACJA UKŁADÓW PRZEŁĄCZALNYCH

4.1. WSTĘP DO ROZDZIAŁU

Poprzednie rozdziały stanowiły wprowadzenie do bezpośredniej metody Lapunowa, która jest formalnym narzędziem analizy stabilności dynamiki układów. W odniesieniu do układów o niecałkowitych rzędach różniczkowania przedstawiono podstawowe aspekty ich opisu oraz zaproponowano autorskie podejścia do konstrukcji funkcji Lapunowa dla liniowych układów ułamkowego rzędu. Należy jednak zauważyć, że w przypadku układów liniowych stabilność może być badana na podstawie istniejących kryteriów analitycznych, które eliminują konieczność bezpośrednią konstrukcji funkcji Lapunowa. Głównym celem analiz dotyczących układów liniowych w niniejszej pracy było rozwiązywanie fundamentalnych problemów związanych z określeniem funkcji Lapunowa, co umożliwia jej zastosowanie do bardziej złożonych klas układów, w których standardowe kryteria stabilności mogą być niewystarczające. Szczególną kategorią takich układów są układy nieliniowe oraz układy przełączalne. W bieżącym rozdziale zostanie przeprowadzona analiza stabilności specyficznej klasy układów przełączalnych, które nie oddają się klasycznym metodom oceny stabilności, a następnie zaproponowane zostaną metody stabilizacji problemowego układu w oparciu o metodę Lapunowa.

4.2. UKŁADY PRZEŁĄCZALNE

Układy przełączalne są szczególnym przypadkiem dynamiki niestacjonarnej. Parametry modelu zmieniają się według funkcji przełączeń, która dyktuje, w jaki sposób i kiedy dochodzi do zmian w układzie. D. Liberzon w [49] szczegółowo omawia układy przełączalne, definiuje sposób ich opisu, sposoby badania stabilności oraz inne rzeczy. Obecny podrozdział zostanie poświęcona wyłącznie podstawom modelowania układów przełączalnych. W ogólnym wypadku układy przełączalne całkowitego rzędu są opisywane równaniem:

$$\dot{x}(t) = f_{\sigma(t)}(x(t)), \quad (4.1)$$

gdzie:

- $\sigma(t)$ – funkcja przełączeń.

Jeżeli układy są liniowe, w przypadku autonomicznym równanie upraszcza się do:

$$\dot{x}(t) = A_{\sigma(t)}x(t), \quad (4.2)$$

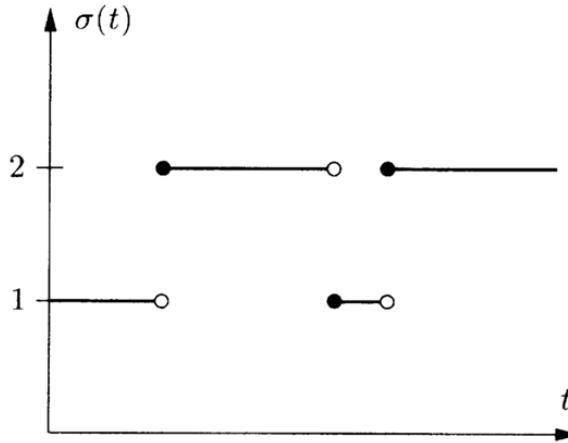
gdzie:

- $A_{\sigma(t)}$ – macierz stanu dla określonej wartości funkcji $\sigma(t)$.

Natomiast jeżeli układ zostaje poddany zewnętrznemu sterowaniu, równanie można rozbudować:

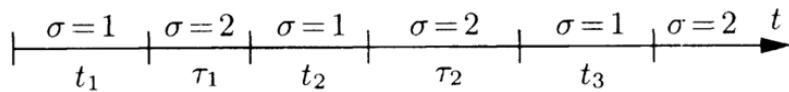
$$\dot{x}(t) = A_{\sigma(t)}x(t) + Bu(t). \quad (4.3)$$

Rysunek 4.1 przedstawia przykładową funkcję przełączeń, która mówi, że jeżeli jej wartość wynosi $\sigma(t) = 1$, wtedy dynamikę układu opisuje macierz stanu A_1 . W wyniku zmian funkcji przełączeń zmienia się dynamika układu.



Rysunek 4.1: Przykładowa funkcja przełączeń. Źródło: [49] str. 7.

Odnosząc się do stabilności, w wypadku liniowych układów przełączalnych całkowitego rzędu, powinno być wystarczające stwierdzenie, że w większości przypadków, jeśli układy, które składają się na przełączenie, są stabilne, wtedy układ przełączalny powinien być stabilny i w większości przypadków faktycznie tak jest. Takie rozumowanie wynika z faktu, że jeśli dany jest sposób przełączeń z Rysunku 4.2, wtedy odpowiedź liniowego, autonomicznego układu



Rysunek 4.2: Opis przełączeń układu. Źródło: [49], str. 31.

przełączalnego jest dana jako:

$$x(t) = \dots e^{A_2 \tau_2} e^{A_1 t_2} e^{A_2 \tau_1} e^{A_1 t_1} x_0. \quad (4.4)$$

Mogłoby się wydawać, że jeśli podukłady są stabilne, wtedy odpowiedź układu z warunku początkowego powinna dążyć do zera. Jednak dzieje się tak wyłącznie w wypadku układów komutacyjnych, które spełniają warunek:

$$[A, B] = AB - BA = 0 \rightarrow e^A e^B = e^{A+B}, \quad (4.5)$$

ponieważ wtedy (4.4) można zapisać jako:

$$x(t) = e^{A_2(\tau_1 + \tau_2 + \dots)} e^{A_1(t_1 + t_2 + \dots)}, \quad (4.6)$$

czego dowód można znaleźć w [49] str. 31.

4.3. PROBLEM STABILIZACJI

Istnieją liniowe układy przełączalne niespełniające warunku (4.5), których podukłady są stabilne, natomiast ich ogólna stabilność zależy głównie od funkcji przełączeń.

PRZYKŁAD 4.1: Dane są macierze stanu:

$$A_1 = \begin{bmatrix} -0.1 & -1 \\ 2 & -0.1 \end{bmatrix}, \quad A_2 = \begin{bmatrix} -0.1 & 2 \\ -1 & -0.1 \end{bmatrix}. \quad (4.7)$$

Układ opisany jest równaniem:

$$\dot{x}(t) = \begin{cases} A_1 x(t) & \text{if } \sigma(t) = 1, \\ A_2 x(t) & \text{if } \sigma(t) = 2. \end{cases} \quad (4.8)$$

Macierze stanu nie spełniają warunku (4.5), ponieważ:

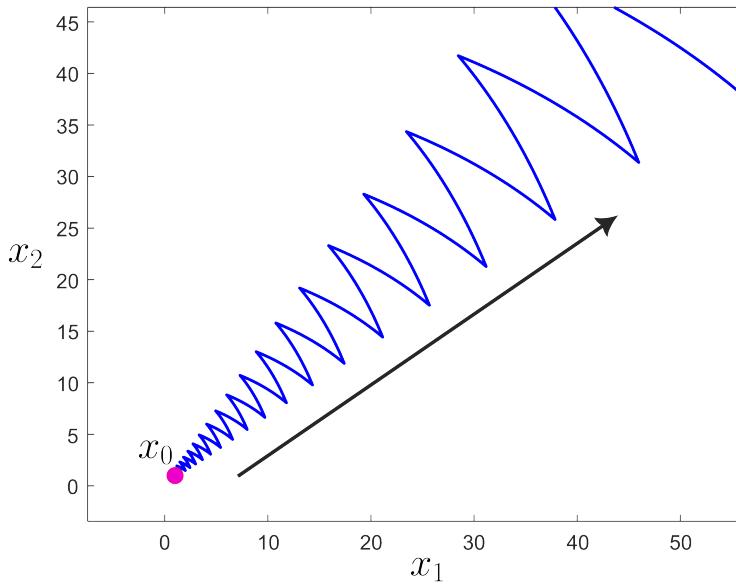
$$[A_1, A_2] = \begin{bmatrix} -0.1 & -1 \\ 2 & -0.1 \end{bmatrix} \begin{bmatrix} -0.1 & 2 \\ -1 & -0.1 \end{bmatrix} - \begin{bmatrix} -0.1 & 2 \\ -1 & -0.1 \end{bmatrix} \begin{bmatrix} -0.1 & -1 \\ 2 & -0.1 \end{bmatrix} = \begin{bmatrix} -3 & 0 \\ 0 & 3 \end{bmatrix}. \quad (4.9)$$

Bieguny podukładów wynoszą odpowiednio:

$$\lambda(A_1) = \begin{cases} \lambda_1 = -0.1 + \sqrt{2}i, \\ \lambda_2 = -0.1 - \sqrt{2}i, \end{cases} \quad (4.10)$$

$$\lambda(A_2) = \begin{cases} \lambda_1 = -0.1 + \sqrt{2}i, \\ \lambda_2 = -0.1 - \sqrt{2}i \end{cases}$$

i znajdują się w lewej półotwartej półpłaszczyźnie zespolonej, więc są stabilne. Jednak jeśli funkcją przełączającą jest sygnałem prostokątnym o współczynniku wypełnienia $D = 50\%$ i okresie $T = 0.5 \text{ s}$, wtedy autonomiczny układ przełączalny z warunkiem początkowym $x_0 = [1, 1]^T$, nie jest stabilny, co przedstawia Rysunek 4.3. Istnieją funkcje przełączeń, będące sygnałem prostokątnym, z których część stabilizuje układ, natomiast część sprawia, że jest on niestabilny. Problemem jest opracowanie algorytmu, który automatycznie zadaje funkcję przełączeń, która stabilizuje system. Taka potrzeba wynika z faktu, że gdyby funkcja przełączeń była ustalana arbitralnie, wtedy dla każdego układu, i drobnych zmian w parametrach, które



Rysunek 4.3: Niestabilna trajektoria układu przełączalnego.

mogą zachodzić wraz z czasem trzeba byłoby wykonywać obliczenia od początku. Natomiast algorytm automatycznej stabilizacji jest w stanie automatycznie dobrać odpowiednią funkcję przełączeń, dla dowolnych, problemowych macierzy stanu.

UWAGA 4.1: Na początku obecnego 2025 r. został opublikowany prawidłowy warunek badania stabilności układów przełączalnych całkowitego rzędu, który jednoznacznie określa dlaczego część funkcji przełączeń stabilizują układ, a część nie [50]. Warunek przybiera postać:

$$\prod_{j=1}^N \|e^{A_j}\|^{c_j} < 1, \quad (4.11)$$

gdzie:

- N – liczba podukładów,
- c – współczynnik określający stosunek czasu działania układu do całości okresu funkcji przełączeń.
- $\|A\|_\infty = \max_{1 \leq i \leq m} \sum_{j=1}^n |a_{ij}|$

W przedstawionym przypadku współczynniki są sobie równe i wynoszą $c = 0.5$. Po podstawieniu danych i obliczeniu eksponenty macierzowej:

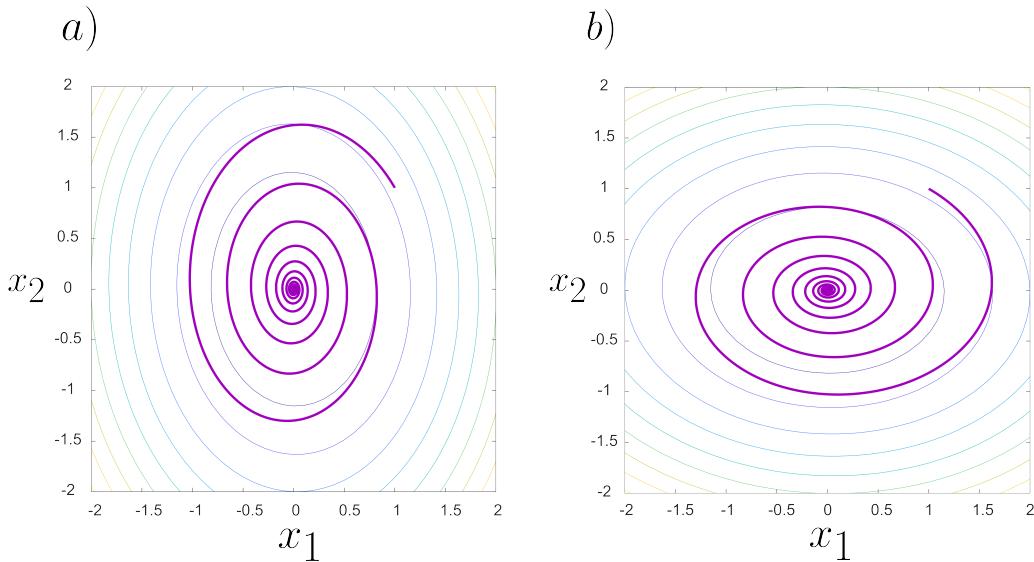
$$\left\| \begin{bmatrix} 0.1411 & -0.6320 \\ 1.2640 & 0.1411 \end{bmatrix} \right\|^{0.5} \left\| \begin{bmatrix} 0.1411 & 1.2640 \\ -0.6320 & 0.1411 \end{bmatrix} \right\|^{0.5} = 1.4051^{0.5} \cdot 1.4051^{0.5} = 1.4051, \quad (4.12)$$

co jest większe od jedności i nie spełnia warunku stabilności. Z tego względu Rysunek 4.3 przedstawia układ przełączalny, który nie jest stabilny. Metoda opisywana [50] pozwala na ana-

lityczne wyznaczenie funkcji przełączeń, która stabilizuje układ. Obecna praca nad funkcjami Lapunowa i metoda [50] są w pewnym zakresie ze sobą tożsame. Algorytm opracowany w następnych sekcjach automatycznie dokonuje syntezy funkcji przełączeń, która spełnia warunek (4.11). Ponadto algorytm ma zastosowanie również do układów o rzędzie niecałkowitym.

4.4. OPRACOWANIE ALGORYTMU STABILIZACJI

Obecna sekcja opisuje badanie stabilności problemowego układu, przy użyciu metody Lapunowa. Metoda została wybrana ze względu na brak innej możliwości badawczej stabilności problemowego układu, która dodatkowo pozwoli na opracowanie algorytmu stabilizacji. Rysunek 4.4 przedstawia trajektorie podukładów z warunku początkowego $x_0 = [1, 1]^T$ wraz z poziomicami funkcji Lapunowa, które zostały dobrane zgodnie z założeniami Algebraicznego Równania Riccatiego, przedstawionymi w poprzednich rozdziałach. Z przedstawionych poziomic funkcji Lapunowa oraz trajektorii podukładów można wnioskować, że podczas zwrotu wirowania pola wektorowego możliwa jest sytuacja, w której przełączenie będzie powodowało zwiększenie wartości funkcji Lapunowa po przełączeniu. Jeżeli do wzrostu wartości funkcji będzie dochodziło cyklicznie, wtedy układ przełączalny nie będzie stabilny. W celu wykonania stabilizacji wystarczy zadbać o to, żeby przełączenie układów następowało w momentach, które gwarantują zmniejszenie wartości funkcji Lapunowa po przełączeniu, a co za tym idzie zmniejszenia energii układu, do osiągnięcia założonego stanu ustalonego. Podczas działania



Rysunek 4.4: Trajektorie podukładów wraz z poziomicami funkcji Lapunowa; a) parametry z macierzy A_1 ; b) parametry z macierzy A_2 .

jednego układu trzeba weryfikować wartości drugiej funkcji, tak żeby do przełączenia potencjalnie zmniejszały ogólną energię, stabilizując układ. Z przedstawionego względzu wynika, że

można zapisać funkcje permutacji dostępnych macierzy P :

$$\begin{cases} V_{12} = (x - x_{e_1})^T P_2^* (x - x_{e_1}), \\ V_{21} = (x - x_{e_2})^T P_1^* (x - x_{e_2}), \end{cases} \quad (4.13)$$

gdzie:

- P^* – znormalizowana macierz parametrów funkcji Lapunowa.

Normalizacja występuje w celu porównania dwóch funkcji, które mogą mieć różne wartości. W przedstawianej metodzie została wykorzystana normalizacja według normy macierzowej Frobeniusa:

$$\begin{cases} P_1^* = \frac{P_1}{\|P_1\|_F}, \\ P_2^* = \frac{P_2}{\|P_2\|_F}. \end{cases} \quad (4.14)$$

Metoda polega na uzyskaniu funkcji przełączeń o zmiennym współczynniku wypełnienia, która będzie stabilizowała układ. Z tego względu Modulacja Szerokości Impulsu (MSI eng. PWM) jest podstawową techniką wykorzystaną do odtworzenia funkcji przełączeń. Uzyskanie odpowiedniego współczynnika wypełnienia odbywa się przez porównanie funkcji Lapunowa i następnie ograniczenie współczynnika wypełnienia do wartości z przedziału od 0 do 1, czyli od 0% do 100% okresu fali prostokątnej, zgodnie z równaniem:

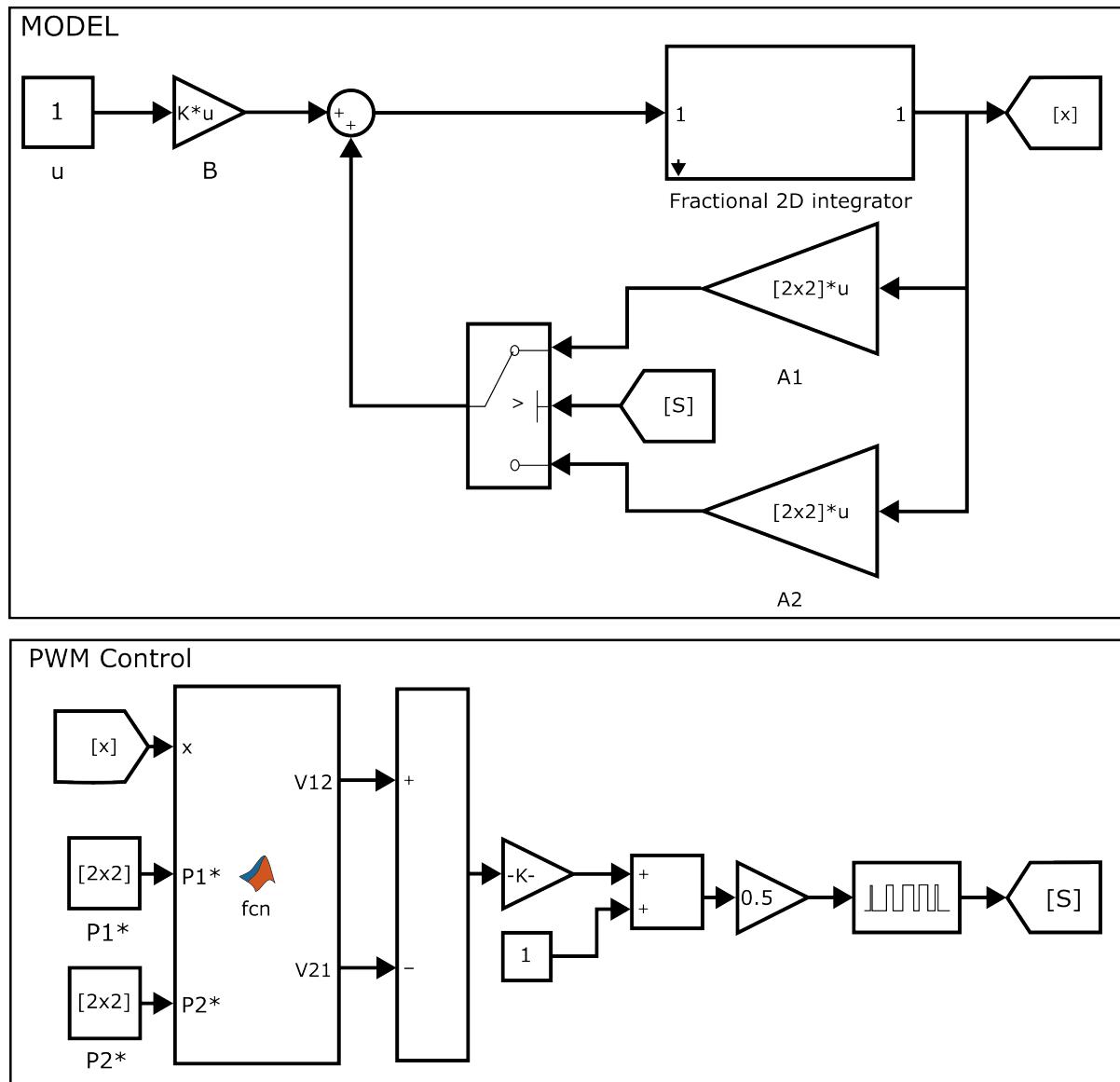
$$D = \left[\frac{V_{12} - V_{21}}{\max(V_{12} - V_{21})} + 1 \right] \cdot \frac{1}{2} \quad (4.15)$$

Rysunek 4.5 przedstawia model układu przełączalnego wraz z algorytmem stabilizacji, który umożliwia badanie jego funkcjonowania również w warunkach obiektu niecałkowitego rzędu. Fractional 2D integrator zawiera elementy biblioteki FOMCON, która umożliwia odtworzenie rachunku różniczkowo-całkowego niecałkowitych rzędów, przy wykorzystaniu aproksymacji Oustaloupa, co zostało opisane w poprzednim rozdziale. Blok `fcn` realizuje obliczenia wartości funkcji Lapunowa, które następnie odtwarzają współczynnik wypełnienia.

4.5. BADANIE ALGORYTMU STABILIZACJI

Badanie opracowanego algorytmu jest przeprowadzone pod kątem stosownalności i wykazania różnic w funkcjonowaniu proponowanych w poprzednim rozdziale metod doboru parametrów funkcji Lapunowa układu liniowego o dynamice ułamkowej. Wyróżnione zostały trzy metody:

- optymalizacja warunkowa,
- optymalizacja funkcjonału kosztu,
- wykorzystanie funkcji Lapunowa całkowitego rzędu.



Rysunek 4.5: Model układu przełączalnego z algorytmem stabilizacji

Optymalizacja warunkowa zwraca parametry funkcji Lapunowa dla układów niecałkowitego rzędów α , zestawione w Tabeli 4.1. P_1 i P_2 są parametrami funkcji Lapunowa uzyskanymi odpowiednio dla macierzy A_1 i A_2 , natomiast wartości d_1 oraz d_2 są współczynnikami kończącymi optymalizację warunkową zgodnie z (3.36). P_1^* i P_2^* są macierzami o wartościach znormalizowanych według normy Frobeniusa. Drugą empiryczną metodą doboru funkcji Lapunowa do dynamiki układu niecałkowitego rzędu jest optymalizacja funkcjonału kosztu. Macierze wynikowe wraz z ich normalizacją zestawia Tabela 4.2. W metodzie, gdzie brane są pod uwagę wyłącznie parametry funkcji Lapunowa całkowitego rzędu, dla rzędów $\alpha = (0, 1]$, brane są pod uwagę wartości macierzy P_1 oraz P_2 uzyskane z metody optymalizacji warunkowej dla $\alpha = 1$. Metodą badawczą, weryfikującą jakość funkcjonowania metod jest przetestowanie ich w algorytmie stabilizacji układu przełączalnego niecałkowitego rzędu 4.5. Model został poddany sterowaniu $u = 1$, przy zerowych warunkach początkowych $x_0 = [0 \ 0]^T$. Okres sygnału

Tabela 4.1: Parametry funkcji Lapunowa według metody optymalizacji warunkowej dla rzędów α

α	P_1	P_2	$\begin{bmatrix} d_1 \\ d_2 \end{bmatrix}$	P_1^*	P_2^*
1.0	7.488 0.124 0.124 3.756	3.756 0.124 0.124 7.488	1.0 1.0	0.895 0.015 0.015 0.449	0.449 0.015 0.015 0.895
0.9	5.803 0.124 0.096 3.756	2.780 0.124 0.920 7.487	0.775 0.740	0.839 0.017 0.017 0.543	0.346 0.015 0.114 0.931
0.8	5.120 0.124 0.085 3.756	2.423 0.124 0.080 7.487	0.684 0.645	0.806 0.020 0.013 0.591	0.308 0.016 0.010 0.951
0.7	4.569 0.124 0.076 3.756	2.122 0.124 0.070 7.487	0.609 0.565	0.772 0.021 0.013 0.635	0.273 0.016 0.009 0.962
0.6	4.073 0.124 0.068 3.756	1.878 0.124 0.062 7.487	0.544 0.500	0.735 0.022 0.012 0.678	0.243 0.016 0.008 0.970
0.5	3.661 0.124 0.061 3.756	1.679 0.124 0.056 7.487	0.489 0.447	0.698 0.024 0.012 0.716	0.219 0.016 0.007 0.976

MSI wynosi 0.001 s , krok dyskretyzacji modelu jest równy 10^{-5} s , natomiast całkowity czas symulacji to 40 s . Z modelu uzyskiwane są dwa sygnały x_1 oraz x_2 . Każdy z sygnałów jest przetworzony na całkowy wskaźnik jakości sygnału:

$$IAE = \sum_0^k [|x_e - x(k)|\Delta T]. \quad (4.16)$$

Odpowiednio IAE_1 oraz IAE_2 , dla x_1 i x_2 . Tabela 4.3 zestawia wyniki przeprowadzonych badań. Została podzielona w taki sposób, że pierwszą kolumnę stanowi rząd układu. Kolumna *OW* odpowiada metodzie optymalizacji warunkowej, *OFK* oznacza metodę optymalizacji funkcjonału kosztu, natomiast *BO* jest brakiem optymalizacji - metodą, w której dobierane są wyłącznie parametry funkcji Lapunowa rzędu całkowitego, które spełniają warunki funkcji również dla rzędów mniejszych od jedności. W każdej metodzie znajdują się trzy kolumny. Dwie z nich odpowiadają całkowitym wskaźnikom jakości otrzymanych sygnałów, natomiast trzecia jest sumą dwóch poprzednich. Ostatni wiersz tabeli jest sumą wszystkich poprzednich wierszy. Sumarycznie wszystkie wyniki wskazują na to, że optymalizacja warunkowa jest najmniej pożądaną metodą doboru funkcji Lapunowa jeśli chodzi o rzędy układów mniejszych od jedności. Posiada jednak możliwość jej rozwinięcia do układów o rzędach większych od jedności. Optymalizacja funkcjonału kosztu okazała się być najlepszą metodą, która ma zastosowanie

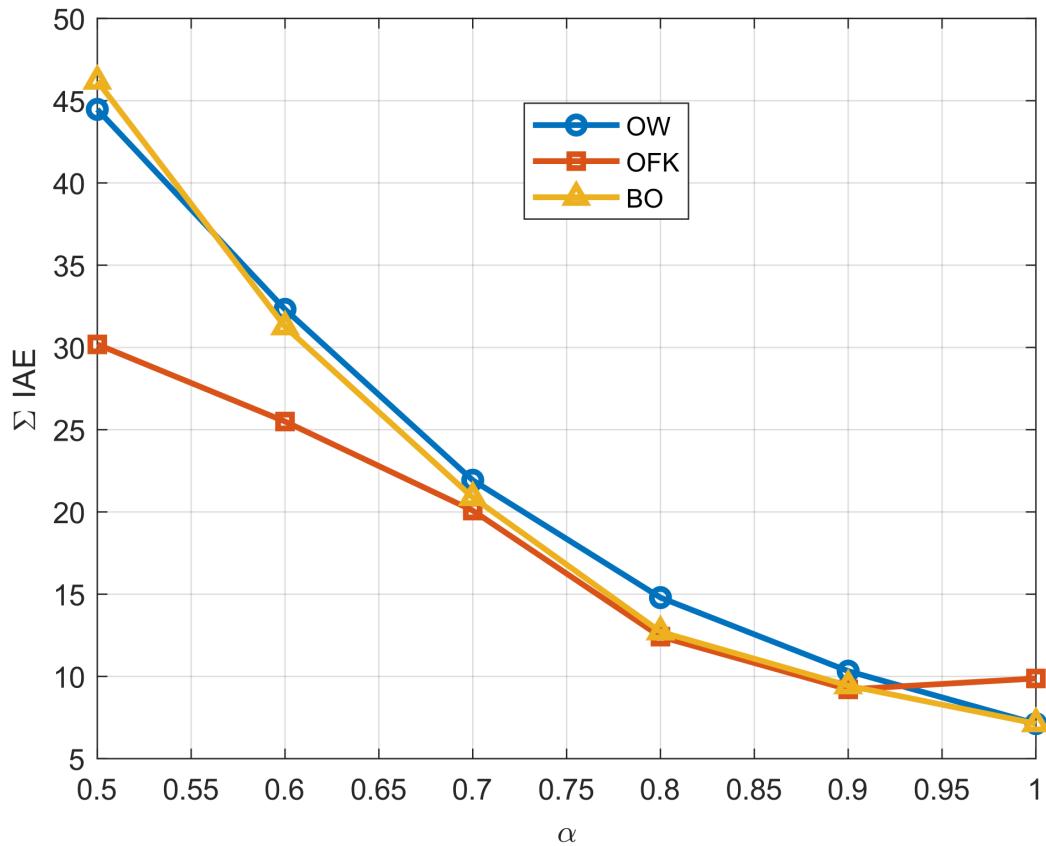
Tabela 4.2: Macierze P_1 i P_2 oraz ich znormalizowane odpowiedniki uzyskane w metodzie optymalizacji funkcjonału kosztu.

α	P_1	P_2	P_1^*	P_2^*
1.0	$\begin{bmatrix} 6.643 & 0.061 \\ 0.162 & 3.332 \end{bmatrix}$	$\begin{bmatrix} 3.338 & 0.355 \\ -0.132 & 6.650 \end{bmatrix}$	$\begin{bmatrix} 0.8936 & 0.0082 \\ 0.0218 & 0.4482 \end{bmatrix}$	$\begin{bmatrix} 0.4480 & 0.0476 \\ -0.0177 & 0.8926 \end{bmatrix}$
0.9	$\begin{bmatrix} 2.367 & 0.098 \\ 0.072 & 1.085 \end{bmatrix}$	$\begin{bmatrix} 1.168 & 0.362 \\ -0.114 & 2.014 \end{bmatrix}$	$\begin{bmatrix} 0.9081 & 0.0376 \\ 0.0275 & 0.4163 \end{bmatrix}$	$\begin{bmatrix} 0.4953 & 0.1535 \\ -0.0482 & 0.8537 \end{bmatrix}$
0.8	$\begin{bmatrix} 1.592 & -0.029 \\ 0.118 & 0.697 \end{bmatrix}$	$\begin{bmatrix} 0.755 & 0.009 \\ 0.237 & 1.093 \end{bmatrix}$	$\begin{bmatrix} 0.9137 & -0.0168 \\ 0.0679 & 0.4003 \end{bmatrix}$	$\begin{bmatrix} 0.5598 & 0.0068 \\ 0.1753 & 0.8098 \end{bmatrix}$
0.7	$\begin{bmatrix} 1.289 & 0.094 \\ -0.051 & 0.620 \end{bmatrix}$	$\begin{bmatrix} 0.582 & -0.160 \\ 0.330 & 0.753 \end{bmatrix}$	$\begin{bmatrix} 0.8985 & 0.0657 \\ -0.0357 & 0.4325 \end{bmatrix}$	$\begin{bmatrix} 0.5707 & -0.1566 \\ 0.3235 & 0.7384 \end{bmatrix}$
0.6	$\begin{bmatrix} 1.145 & -0.099 \\ 0.138 & 0.683 \end{bmatrix}$	$\begin{bmatrix} 0.505 & 0.125 \\ -0.043 & 0.729 \end{bmatrix}$	$\begin{bmatrix} 0.8520 & -0.0733 \\ 0.1028 & 0.5080 \end{bmatrix}$	$\begin{bmatrix} 0.5631 & 0.1388 \\ -0.0481 & 0.8133 \end{bmatrix}$
0.5	$\begin{bmatrix} 1.147 & 0.016 \\ 0.022 & 0.941 \end{bmatrix}$	$\begin{bmatrix} 0.483 & -0.018 \\ -0.017 & 0.993 \end{bmatrix}$	$\begin{bmatrix} 0.7730 & 0.0107 \\ 0.0148 & 0.6342 \end{bmatrix}$	$\begin{bmatrix} 0.4372 & -0.0163 \\ -0.0151 & 0.8991 \end{bmatrix}$

w wypadku implementacji w algorytmie stabilizacji. Odpowiedzi wszystkich układów zachowują stabilność w kontekście przedstawionych rzędów. Rysunek 4.6 zestawia wyniki zawarte w Tabeli 4.3. Można dostrzec zależność, która wskazuje, że podczas zmniejszania rzędu układu lepiej sprawuje się metoda optymalizacji funkcjonału kosztu. Jednak w wypadku rzędów od 0.7 do 0.9 wyniki wszystkich metod są do siebie zbliżone.

Tabela 4.3: Wskaźniki całkowe otrzymane z modelu, dla różnych rzędów α

α	OW			OFK			BO		
	IAE_1	IAE_2	\sum	IAE_1	IAE_2	\sum	IAE_1	IAE_2	\sum
1	2.718	4.413	7.131	3.152	6.725	9.877	2.718	4.413	7.131
0.9	4.023	6.300	10.323	4.353	4.860	9.213	4.419	5.019	9.438
0.8	5.931	8.861	14.792	6.628	5.791	12.419	6.235	6.478	12.713
0.7	8.545	13.382	21.927	10.958	9.125	20.083	8.710	12.173	20.883
0.6	11.885	20.430	32.315	13.979	11.517	25.496	11.130	20.130	31.260
0.5	16.560	27.911	44.471	10.144	20.040	30.184	15.695	30.495	46.190
\sum	49.662	81.297	130.959	49.213	58.058	107.272	49.907	78.708	127.562



Rysunek 4.6: Porównanie sumy wskaźników jakości dla różnych rzędów układu

4.6. PODSUMOWANIE ROZDZIAŁU

Rozdział ten przedstawia opracowaną metodę automatycznej stabilizacji układów przełączalnych, której głównym założeniem jest wykorzystanie funkcji Lapunowa do dynamicznej oceny stabilności układu w trakcie jego działania. Proponowane podejście zakłada konstrukcję

reguły przełączania opartej na ocenie wartości funkcji Lapunowa dla poszczególnych podukładów, co w założeniu ma prowadzić do wyboru takiej trajektorii przełączeń, która zapewnia stabilność całego układu.

Należy jednak zaznaczyć, że metoda ta na obecnym etapie nie posiada formalnego dowodu poprawności – jej skuteczność została potwierdzona jedynie poprzez eksperymenty numeryczne zaprezentowane w ramach niniejszego rozdziału. Tym samym jej wiarygodność opiera się wyłącznie na wynikach obliczeń symulacyjnych, co stanowi istotne ograniczenie, zwłaszcza z perspektywy dalszego wykorzystania tej metody w zastosowaniach inżynierskich lub naukowych wymagających rygorystycznego ugruntowania matematycznego.

Opracowane w poprzednich rozdziałach metody syntezy stacjonarnych funkcji Lapunowa — zarówno podejścia analityczne, jak i bazujące na numerycznej minimalizacji funkcjonału kosztu — umożliwiły rozszerzenie zastosowania algorytmu stabilizacji na układy dynamiczne opisywane równaniami różniczkowymi niecałkowitego rzędu. W tym ujęciu, zastosowanie tych metod pozwoliło na konstrukcję kryterium przełączeń również w kontekście bardziej złożonych modeli fizycznych, w których klasyczne metody stabilizacji okazują się niewystarczające.

Główne ograniczenia zaproponowanego algorytmu wynikają z dwóch zasadniczych kwestii. Po pierwsze, jak już wspomniano, brak jest matematycznego dowodu poprawności działania algorytmu, co uniemożliwia jednoznaczne określenie jego zakresu stosownalności. Po drugie, funkcjonowanie algorytmu jest silnie zależne od przyjętej metody syntezy funkcji Lapunowa, której skuteczność i zakres stosownalności są ograniczone przez przyjęte założenia oraz charakterystyki analizowanego układu dynamicznego. Z tego względu nie można jednoznacznie określić, w jakim stopniu algorytm ten może być uogólniony na inne klasy układów przełączalnych, zwłaszcza o nietypowej dynamice czy niestandardowych strukturach modeli.

5. PODSUMOWANIE PRACY

Drugi rozdział pracy stanowi teoretyczną podstawę, niezbędną w dalszej analizie stabilności układów dynamicznych przy użyciu bezpośredniej metody Lapunowa. W sposób metodyczny wprowadza czytelnika w podstawowe pojęcia związane z teorią układów dynamicznych oraz przybliża klasyczne modele matematyczne służące do ich opisu. Przedstawione zagadnienia oraz przykłady zostały dobrane w taki sposób, aby stworzyć spójną i logiczną narrację, prowadzącą od elementarnych definicji aż po bardziej złożone aspekty teoretyczne niezbędne do pełnego zrozumienia dalszych rozważań i analiz. W rozdziale omówiono także podstawowe modele układów dynamicznych całkowitego rzędu, co pozwala na uchwycenie ich właściwości w sposób zwarty i analitycznie przejrzysty. Szczególną uwagę poświęcono liniowym układom dynamicznym, które stanowią naturalny punkt wyjścia dla rozważań nad układami bardziej złożonymi, takimi jak układy przełączalne czy o niecałkowitym rzędzie. Przedstawione modele zostały opisane z wykorzystaniem różnych formalizmów, co umożliwia wieloaspektową interpretację zachowania układów. Podkreślono również związek między opisem matematycznym a fizycznymi właściwościami układów, w tym szczególnie związków z całkowitą energią układu, Hamiltonianem i funkcjami Lapunowa. Omówiono znaczenie optymalizacji w analitycznym procesie doboru funkcji Lapunowa układów liniowych. Dzięki temu rozdział ten nie tylko przygotowuje grunt pod pozostałe części pracy, ale także ukazuje fundamentalne powiązania między teorią układów dynamicznych a klasyczną fizyką teoretyczną. Całość stanowi wprowadzenie w problematykę stabilności układów dynamicznych, a także stanowi pierwszą część teoretycznego zakresu pracy, skoncentrowanego wokół bezpośredniej metody Lapunowa.

Rozdział trzeci pracy poświęcony został wprowadzeniu rachunku różniczkowego niecałkowitego rzędu do opisu dynamiki układów, w których tradycyjny formalizm oparty na klasycznych pochodnych okazuje się niewystarczający do wiernego odwzorowania obserwowanych zjawisk. W szczególności omówiono podstawowe definicje operatorów różniczkowych ułamkowego rzędu, takich jak pochodna w sensie Caputo, Riemanna-Liouville'a czy Grunwalda-Letnikowa, wraz z ich właściwościami i ograniczeniami. Integralną częścią rozdziału jest przegląd istniejących bibliotek środowiska MATLAB, umożliwiających modelowanie dynamiki układów niecałkowitego rzędu. Zaprezentowano zarówno gotowe funkcje dostępne w ramach dostępnych toolboxów, jak i rozwiązania autora pracy, które wykorzystywane są w badaniach eksperymentalnych i symulacjach numerycznych. Szczególny nacisk położono na analizę warunków początkowych, których interpretacja i zastosowanie w kontekście modeli niecałkowitego rzędu budzi istotne kontrowersje. Przeprowadzona analiza wykazała, że wybór definicji pochodnej ułamkowej bezpośrednio wpływa na sposób formułowania i interpretacji warunków początkowych, co może prowadzić do znacznych różnic w trajektoriach wektora stanu, nawet przy identycznych parametrach układu. W konsekwencji pojawia się trudność w jednoznacznym do-

pasowaniu modelu do rzeczywistego zjawiska fizycznego, co ogranicza praktyczną użyteczność tych modeli w zastosowaniach inżynierskich. W dalszej części rozdziału przedstawiono analizę współczynników w układach o rzędach pochodnych mniejszych od jedności. Rozważania te prowadzą do istotnego wniosku, że możliwe jest zastosowanie klasycznego algebraicznego równania Riccatiego (ARE), znanego z teorii sterowania układów całkowitego rzędu, do syntezy funkcji Lapunowa dla układów o niecałkowitym rzędzie. W ten sposób zarysowano pierwszą z metod syntezy funkcji Lapunowa dla tego typu układów. Innym podejściem, polegającym na minimalizacji funkcjonału kosztu z empirycznie lub symulacyjnie uzyskanych trajektorii układu niecałkowitego rzędu, otwiera drogę do konstrukcji funkcji Lapunowa poprzez optymalizację numeryczną. Zaproponowane podejście stanowi próbę przezwyciężenia trudności związanych z wyprowadzeniem analitycznego rozwiązania, w kontekście układów opisanych rachunkiem różniczkowym ułamkowego rzędu. Przedstawiona optymalizacja prowadzi do uzyskania parametrów funkcji Lapunowa i jest drugą metodą syntezy funkcji Lapunowa przedstawianą w ramach pracy. W dalszej części rozdziału przeprowadzono analizę wpływu struktury funkcji Lapunowa na możliwość zapewnienia warunku ujemności jej pochodnej wzduż trajektorii układu. Szczególną uwagę poświęcono możliwości modyfikacji parametrów funkcji Lapunowa z wykorzystaniem iloczynu Hadamarda, czyli operacji iloczynu element po elemencie macierzy. Tego typu modyfikacja pozwala na subtelną, lokalną regulację kształtu poziomów funkcji Lapunowa, co umożliwia dostosowanie ich do konkretnego charakteru dynamiki układu, przy jednoczesnym zachowaniu podstawowych własności funkcji kandydującej na funkcję Lapunowa, takich jak dodatnia określoność. Wykazano, że odpowiednio dobrany iloczyn Hadamarda pozwala deformować poziomice funkcji Lapunowa w taki sposób, aby w granicy zapewnić spełnienie warunku negatywności jej pochodnej względem zmiennych stanu. Otwiera to możliwość wykorzystania podejścia iteracyjnego lub adaptacyjnego w procesie syntezy funkcji Lapunowa, w którym kształt funkcji jest dynamicznie dopasowywany do trajektorii układu, z zachowaniem wymaganego kierunku zmian. Metoda ta nie posiada jednak formalnego dowodu, który może określić poprawność i zakres stosowalności. Końcowa część rozdziału poświęcona została analitycznej analizie procesu minimalizacji funkcjonału kosztu, stanowiącego podstawę dla syntezy funkcji Lapunowa w układach niecałkowitego rzędu. Przeprowadzone rozważania doprowadziły do interesującego i istotnego wniosku, że parametry funkcji Lapunowa nie są stałe, lecz wykazują zależność od przedziału czasowego, w którym prowadzona jest analiza. Oznacza to, że funkcja Lapunowa przyjmuje postać funkcji zależnej od narastającego okna całkowania, co bezpośrednio wiąże się z charakterem dynamiki układów niecałkowitego rzędu oraz specyfiką samego rachunku ułamkowego. Zastosowana aproksymacja asymptotyczna, pozwalająca na uproszczenie opisu oraz istotnie ogranicza zakres stosowalności metody. O ile rozwiązanie to daje użyteczne wyniki w przypadku układów o rzędach mniejszych od jedności, to jednak analiza wykazuje, że metoda traci swoją spójność i dokładność dla wartości rzędu poniżej $\frac{1}{2}$. Dodatkowo, w wyniku zależności parametrów od czasu, zauważa się, że wartości funkcji Lapunowa w początkowym czasie są znikome lub zerowe (wartość w zerze to NAN) nieza-

leżnie od przyjętych warunków początkowych. Zjawisko to wynika bezpośrednio z obecności składników proporcjonalnych do czasu oraz z osobliwości zastosowanej aproksymacji, która w punkcie zerowym posiada asymptotę. Taki charakter funkcji skutkuje tym, że metoda nie nadaje się do bezpośredniego zaimplementowania w czasie rzeczywistym lub w algorytmach wymagających szybkiej reakcji, gdyż funkcje potrzebują odpowiednio długiego horyzontu czasowego, aby osiągnąć dokładność pozwalającą na skutecną ocenę stabilności. Podsumowując, przedstawiona metoda ma potencjał teoretyczny i może znaleźć zastosowanie w analizie długoterminowej dynamiki układów niecałkowitego rzędu, jednak wymaga dalszych modyfikacji i adaptacji, aby mogła zostać zaimplementowana w praktycznych algorytmach sterowania lub analizy stabilności w czasie rzeczywistym. Szczególnie istotnym kierunkiem dalszych badań wydaje się opracowanie strategii modyfikacji aproksymacji początkowych oraz wprowadzenie korekt kompensujących nieciągłości i osobliwości w punkcie zerowym.

Ostatni rozdział pracy poświęcony został analizie zagadnienia stabilizacji układów przełączalnych, w których dynamika opisywana jest przez zestaw podukładów przełączanych według określonego prawa. Zwrócono uwagę na istotny problem praktyczny i teoretyczny, polegający na tym, że nawet jeśli każdy z podukładów z osobna jest stabilny, to nieodpowiednio dobrane reguły przełączeń mogą prowadzić do niestabilności całego układu. W literaturze znane są już podejścia pozwalające na wyznaczenie warunków stabilności układów przełączalnych dla arbitralnie zadanych reguł przełączania, jednak ich stosowalność jest ograniczona ze względu na konieczność ścisłego określenia funkcji przełączeń. W pracy zaproponowano alternatywne podejście w postaci algorytmu automatycznej stabilizacji układów przełączalnych, opartego na bieżącej ocenie wartości funkcji Lapunowa. Kluczową cechą tego algorytmu jest dynamiczne dostosowywanie reguły przełączeń w taki sposób, aby w każdej chwili zapewnić malejącą wartość funkcji Lapunowa, co w praktyce przekłada się na asymptotyczną stabilność całego układu. Chociaż brak jest formalnego dowodu zbieżności algorytmu w ogólności, to w ramach przeprowadzonych badań symulacyjnych algorytm konsekwentnie prowadził do stabilizacji układów, co stanowi silny argument przemawiający za jego skutecznością. Dzięki wcześniejszym analizom i opracowaniu metod syntezy funkcji Lapunowa możliwe było rozszerzenie działania algorytmu na klasę układów przełączalnych o niecałkowitych rzędach. W ten sposób potwierdzono jego uniwersalność oraz możliwość adaptacji do bardziej złożonych modeli dynamicznych. Ponadto, sam algorytm automatycznej stabilizacji został wykorzystany jako narzędzie oceny efektywności różnych metod syntezy funkcji Lapunowa. Analiza wyników wykazała, że metoda oparta na numerycznej optymalizacji funkcjonału kosztu wykazuje najwyższą skuteczność spośród badanych podejść.

Pomimo przedstawienia szeregu oryginalnych i potencjalnie wartościowych metod syntezy funkcji Lapunowa dla układów niecałkowitego rzędu, praca posiada również istotne ograniczenia. Główną słabością opracowanych metod jest brak formalnych dowodów matematycznych potwierdzających ich poprawność oraz ogólną zbieżność. Zaproponowane techniki, w tym algorytm automatycznej stabilizacji oraz metody syntezy funkcji Lapunowa oparte na minima-

lizacji funkcjonału kosztu, zostały zweryfikowane wyłącznie na drodze symulacyjnej. Choć wyniki tych testów są obiecujące, brak dowodów analitycznych ogranicza możliwość ich zastosowania w kontekście teoretycznych badań lub w praktykach wymagających gwarantowanej niezawodności. Kolejnym ograniczeniem jest brak formalnego określenia zakresu stosowalności opracowanych metod, w szczególności względem klasy układów dynamicznych, dla których przedstawione techniki mogą być skutecznie i bezpiecznie zastosowane. W pracy nie wyznaczono jednoznacznych kryteriów, które pozwalałyby ocenić, dla jakich parametrów i struktur układów można bezpiecznie dokonać syntezy funkcji Lapunowa z wykorzystaniem zaproponowanych podejść. Dodatkowo, praca nie zawiera empirycznych ani laboratoryjnych weryfikacji zaprezentowanych koncepcji. Wnioski opierają się wyłącznie na modelach matematycznych oraz symulacjach komputerowych. Brak rzeczywistych danych pomiarowych lub implementacji w fizycznych układach dynamicznych ogranicza wartość aplikacyjną zaproponowanych metod. Pomimo wskazanych ograniczeń, należy podkreślić, że ich pełne wyeliminowanie lub formalne uzupełnienie wykracza poza ramy i założony zakres niniejszej pracy. Wskazane słabości stanowią jednak istotne kierunki dalszych badań i rozwoju przedstawionych metod. Świadomość tych ograniczeń jest istotna zarówno z punktu widzenia dalszych analiz teoretycznych, jak i potencjalnych zastosowań praktycznych, dlatego też zostały one celowo uwzględnione w podsumowaniu.

BIBLIOGRAFIA

- [1] T. Kaczorek, *Teoria Sterowania i Systemów*. Wydawnictwo Naukowe PWN, Wydanie nr 3, ISBN : 83-01-12072-X, Warszawa, Polska, 1993.
<https://search.worldcat.org/title/749574418?oclcNum=749574418>
- [2] S. Bolkowski, *Teoria Obwodów Elektrycznych*. Wydawnictwo Naukowe PWN, Wydanie nr 10, ISBN : 978-83-01-18606-7, Warszawa, Polska, 2020.
<https://search.worldcat.org/title/1192548889?oclcNum=1192548889>
- [3] Norman S. Nise, *Control systems engineering*. John Wiley & Sons, Inc., Hoboken, NJ, 2019.
<https://search.worldcat.org/title/1054262255?oclcNum=1054262255>
- [4] Xue D Y and Bai L, *Fractional Calculus: Numerical Algorithms and Implementations*. Tsinghua University Press, Beijing, 2022.
- [5] F. Rindler, *Calculus of Variations*. Springer Nature, 1st edition, DOI: 10.1007/978-3-319-77637-8, Cham, Switzerland 2018.
<https://link.springer.com/book/10.1007/978-3-319-77637-8>
- [6] MATLAB - *dokumentacja*. Dostęp: 08.12.2024.
<https://www.mathworks.com/help/matlab/>
- [7] NumPy - *dokumentacja*. Dostęp: 16.01.2025.
<https://www.numpy.org/>
- [8] Scipy - *dokumentacja*. Dostęp: 16.01.2025.
<https://www.scipy.org/>
- [9] Python - *dokumentacja*. Dostęp: 16.01.2025.
<https://www.python.org/>
- [10] Matplotlib - *dokumentacja*. Dostęp: 16.01.2025.
<https://www.matplotlib.org/>
- [11] FOTF Toolbox - *dokumentacja*. Dostęp: 20.12.2024.
<https://www.mathworks.com/matlabcentral/fileexchange/60874-fotf-toolbox>
- [12] Curve Fitting Toolbox - *dokumentacja*. Dostęp: 01.01.2025.
<https://www.mathworks.com/help/curvefit/curvefitter-app.html>

- [13] Control System Toolbox - *dokumentacja*. Dostęp: 08.12.2024.
https://www.mathworks.com/help/control/index.html?s_tid=CRUX_lftnav
- [14] Simulink - *dokumentacja*. Dostęp: 20.12.2024.
<https://www.mathworks.com/help/simulink/>
- [15] FOMCON - *dokumentacja*. Dostęp: 20.12.2024.
<https://fomcon.net/>
- [16] C++ - *dokumentacja*. Dostęp: 01.05.2025.
<https://isocpp.org/tour>
- [17] Eigen - *dokumentacja*. Dostęp: 01.05.2025.
https://eigen.tuxfamily.org/index.php?title>Main_Page
- [18] W. Hahn. *Stability of Motion*. Grundlehren Der Mathematischen Wissenschaften, ISBN: 9783540038290, Springer, 2012.
<https://link.springer.com/book/10.1007/978-3-642-50085-5>
- [19] T. Kaczorek. *Selected Problems of Fractional Systems Theory*. Lecture Notes in Control and Information Sciences, DOI: 10.1007/978-3-642-20502-6, Springer, 2011.
<https://link.springer.com/book/10.1007/978-3-642-20502-6+>
- [20] A. F. Fareed, M. S. Semary, H. N. Hassan. *An approximate solution of fractional order Riccati equations based on controlled Picard's method with Atangana–Baleanu fractional derivative*. Alexandria Engineering Journal, Volume 61, Issue 5, Pages 3673-3678, Elsevier B.V., 2022.
<https://doi.org/10.1016/j.aej.2021.09.009>
- [21] R. Sikora. *Pochodne ułamkowe w teorii obwodów elektrycznych Uwagi krytyczne*. Przegląd Elektrotechniczny, Volume 10, Poland, 2016
http://pe.org.pl/abstract_pl.php?nid=10164
- [22] A. Tepljakov, B. B. Alagoz, C. Yeroglu, E. Gonzalez, S. H. HosseinNia, E. Petlenkov. *FOPID Controllers and Their Industrial Applications: A Survey of Recent Results 1*. IFAC-PapersOnLine, Volume 51, Issue 4, Pages 25-30 Elsevier B.V., 2018.
<https://doi.org/10.1016/j.ifacol.2018.06.014>
- [23] E. Pawłuszewicz, A. Koszewnik, P. Burzyński. *On Grünwald-Letinkov Fractional Operator with Measurable Order on Continuous-Discrete Time Scale*. Acta Mechanica et Automatica, Volume 14, Issue 3, Sciendo, 2020.
<https://sciendo.com/article/10.2478/ama-2020-0023>

- [24] G. Tzounas, I. Dassios, M. A. A. Murad, F. Milano. *Theory and Implementation of Fractional Order Controllers for Power System Applications*. IEEE Transactions on Power Systems, Volume 35, Issue 6, Institute of Electrical and Electronics Engineers Inc., 2020.
<https://ieeexplore.ieee.org/document/9106859>
- [25] A. Abd-Elmonem, R. Banerjee, S. Ahmad, W. Jamshed, K. S. Nisar, M. R. Eid, R. W. Ibrahim, S. M. El Din, *A comprehensive review on fractional-order optimal control problem and its solution*. Open Mathematics, Walter de Gruyter GmbH, 2023.
<https://doi.org/10.1515/math-2023-0105>
- [26] A. Koszewnik, E. Pawłuszewicz, M. Ostaszewski, *Experimental Studies of the Fractional PID and TID Controllers for Industrial Process*. International Journal of Control, Automation and Systems, Institute of Control, Robotics and Systems, Volume 19, pages 1847–1862, Springer Nature, 2021.
<https://link.springer.com/article/10.1007/s12555-020-0123-4>
- [27] A. Tepljakov, E. Petlenkov, J. Belikov, *A Flexible MATLAB Tool for Optimal Fractional-order PID Controller Design Subject to Specifications*. Proceedings of the 31st Chinese Control Conference, IEEE, 2012.
<https://ieeexplore.ieee.org/document/6390753>
- [28] T. Kaczorek, K. Rogowski, *Fractional Linear Systems and Electrical Circuits*. Studies in Systems, Decision and Control, Volume 13, Springer International Publishing Switzerland, 2015.
<https://link.springer.com/book/10.1007/978-3-319-11361-6>
- [29] M. Farman, R. Sarwar, A. Akgul, *Modeling and analysis of sustainable approach for dynamics of infections in plant virus with fractal fractional operator*. Chaos, Solitons and Fractals, Volume 170, Elsevier Ltd, 2023.
<https://doi.org/10.1016/j.chaos.2023.113373>
- [30] M. N. Muftah, A. A. M. Faudzi, S. Sahlan, M. Shouran, *Modeling and Fuzzy FOPID Controller Tuned by PSO for Pneumatic Positioning System*. Energies, MDPI, Volume 15, Issue 10, 2022.
<https://www.mdpi.com/1996-1073/15/10/3757>
- [31] R. Rajesh, *Optimal tuning of FOPID controller based on PSO algorithm with reference model for a single conical tank system*. SN Applied Sciences, Volume 1, Springer Nature, 2019.
<https://link.springer.com/article/10.1007/s42452-019-0754-3>
- [32] P. Kumar, D. Baleanu, V. S. Erturk, M. Inc, V. Govindaraj, *A delayed plant disease model with Caputo fractional derivatives*. Advances in Continuous and Discrete Models, Springer

Science and Business Media Deutschland GmbH, 2022.

<https://advancesincontinuousanddiscretemodels.springeropen.com/articles/10.1186/s13662-022-03684-x>

- [33] M. Turkyilmazoglu, M. Altanji, *Fractional models of falling object with linear and quadratic frictional forces considering Caputo derivative*. Chaos, Solitons and Fractals, Volume 166, Elsevier Ltd, 2023.

<https://doi.org/10.1016/j.chaos.2022.112980>

- [34] N. Aguila-Camacho, M. A. Duarte-Mermoud, J. A. Gallegos, *Lyapunov functions for fractional order systems*. Communications in Nonlinear Science and Numerical Simulation, Volume 19, Issue 9, Elsevier B.V., 2014.

<https://doi.org/10.1016/j.cnsns.2014.01.022>

- [35] A. Boukhouima, K. Hattaf, El Mehdi Lotfi, M. Mahrouf, Delfim F.M. Torres, N. Yousfi, *Lyapunov functions for fractional-order systems in biology: Methods and applications*. Chaos, Solitons and Fractals, Volume 140, Elsevier Ltd, 2020.

<https://doi.org/10.1016/j.chaos.2020.110224>

- [36] Zhuo Li, Lu Liu, Sina Dehghan, Yang Quan Chen, Dingyü Xue, *A review and evaluation of numerical tools for fractional calculus and fractional order controls*. International Journal of Control, Taylor and Francis Ltd., 2015.

<https://www.tandfonline.com/doi/full/10.1080/00207179.2015.1124290>

- [37] Marina Popolizio, *On the Matrix Mittag-Leffler Function: Theoretical Properties and Numerical Computation*. Mathematics, Volume 7, Issue 12, MDPI, 2019.

<https://www.mdpi.com/2227-7390/7/12/1140>

- [38] A. Oustaloup, F. Levron, B. Mathieu, F. M. Nanot, *Frequency-band complex noninteger differentiator: characterization and synthesis*. IEEE Transactions on Circuits and Systems I: Fundamental Theory and Applications, Volume 47, Issue 1, IEEE, 2000.

<https://ieeexplore.ieee.org/document/817385>

- [39] J. A. Nelder, R. Mead, *A Simplex Method for Function Minimization*. The Computer Journal, Volume 7, Issue 4, 1965.

<https://doi.org/10.1093/comjnl/7.4.308>

- [40] Lagarias, J. C., J. A. Reeds, M. H. Wright, and P. E. Wright, *Convergence Properties of the Nelder-Mead Simplex Method in Low Dimensions..* SIAM Journal of Optimization, Volume 9, Issue 1, 1998.

<https://doi.org/10.1137/S1052623496303470>

- [41] Jun Fu, Yiwen QI, Ning Xing, Yuzhe Li *A new switching law for event-triggered switched systems under DoS attacks.* Automatica, Volume 142, 2022.
<https://doi.org/10.1016/j.automatica.2022.110373>
- [42] R. Almeida, E. Girejko, L. Machado, A. Malinowska, N. Martins *On the stability of memory-dependent multi-agent systems under DoS attacks.* Journal of The Franklin Institute Engineering and Applied Mathematics, Volume 362, Issue 1, 2025.
<https://doi.org/10.1016/j.jfranklin.2024.107402>
- [43] Rui Zhao, Zhiqiang Zuo, Yijing Wang, Wentao Zhang, *Active control strategy for switched systems against asynchronous DoS attacks.* Automatica, Volume 148, 2023.
<https://doi.org/10.1016/j.automatica.2022.110765>
- [44] R. Zhao, Z. Zuo and Y. Wang *Event-Triggered Control for Switched Systems With Denial-of-Service Attack.* IEEE Transactions on Automatic Control, vol. 67, no. 8, pp. 4077-4090, 2022.
<https://ieeexplore.ieee.org/document/9779527?denied=>
- [45] G. Wu, J. Sun and J. Chen *Optimal Linear Quadratic Regulator of Switched Systems.* IEEE Transactions on Automatic Control, vol. 64, no. 7, pp. 2898-2904, 2019.
<https://ieeexplore.ieee.org/document/8472172>
- [46] Cui-Liu Jin, Rui Wang, Qing-Guo Wang, *Stabilization of switched systems with time-dependent switching signal.* Journal of the Franklin Institute, Volume 357, 2020.
<https://doi.org/10.1016/j.jfranklin.2020.10.007>
- [47] P. Kieczmerski *Switching the dynamics of the Fractional-order system.* 1st International Young Researchers Conference: Ideas Research Training, Białystok, Poland, 2024.
- [48] D. W. Brzeziński, P. Ostalczyk, *The Grünwald-Letnikov formula and its equivalent Horner's form accuracy comparison and evaluation for application to fractional order PID controllers.* 17th International Conference on Methods & Models in Automation & Robotics (MMAR), Miedzyzdroje, Poland, 2012.
<https://ieeexplore.ieee.org/document/6347821>
- [49] D. Liberzon, *Switching in Systems and Control.* Systems & Control: Foundations and Applications, Springer Science+Business Media, Urbana, USA, 2003.
<https://link.springer.com/book/10.1007/978-1-4612-0017-8>
- [50] K. Borawski, *Simple Stability Conditions of Linear Switched Systems with Time-Dependent Switching.* Pomiary Automatyka Robotyka, Nr 1/2025, Polska, 2025.
http://dx.doi.org/10.14313/par_255/55

- [51] F. Mainardi, *Why the Mittag-Leffler Function Can Be Considered the Queen Function of the Fractional Calculus?*. Entropy, 22, 1359, 2020.
<https://doi.org/10.3390/e22121359>