Дніпровський національний університет імені Олеся Гончара
Міністерство освіти і науки України

Дніпровський національний університет імені Олеся Гончара
Міністерство освіти і науки України

Кваліфікаційна наукова праця
на правах рукопису

Кошель Євген Васильович

УДК 004.67, 004.94, 517.9, 534.4

ДИСЕРТАЦІЯ
Моделі та методи синтезу мовлення на основі нейронних систем звичайних диференціальних рівнянь

113 – Прикладна математика

Подається на здобуття ступеня
доктора філософії

Дисертація містить результати власних досліджень. Використання ідей, результатів і текстів інших авторів мають посилання на відповідне джерело.

_____ Кошель Є. В.

Науковий керівник: **Білозьоров Василь Євгенович**
доктор фізико-математичних наук, професор

Дніпро — 2025

Анотація

Кошель Є. В. Моделі та методи синтезу мовлення на основі нейронних систем звичайних диференціальних рівнянь — Кваліфікаційна наукова праця на правах рукопису.

Дисертація на здобуття ступеня доктора філософії за спеціальністю 113 – Прикладна математика. — Дніпровський національний університет імені Олеся Гончара Міністерства освіти і науки України, Дніпро, 2025.

Дисертаційна робота присвячена дослідженню сучасних методів синтезу мовлення на основі новітніх підходів до побудови нейромережевих моделей. Насамперед розглядається новий клас нейромережевих моделей – нейронні звичайні диференціальні рівняння, що дозволяє моделювати динамічні процеси у вигляді систем диференціальних рівнянь, праві частини яких задані нейронними мережами. Головна перевага цього класу моделей у контексті обраної предметної області полягає у тому, що він дозволяє застосовувати відомий інструментарій для дослідження якісних характеристик побудованих моделей.

У ході роботи над дисертацією було розроблено нову модель на базі нейронних звичайних диференціальних рівнянь, що дозволяє апроксимувати динамічні процеси, задані одновимірними часовими рядами, а також адаптуватися до якісних змін цих процесів для збереження якості апроксимації.

Розроблена модель складається з декількох компонентів, які виконують в її рамках різні задачі. Перший такий компонент – модель-кодувальник, що перетворює сегменти часового ряду на точку в просторі вкладення. Цей компонент дозволяє переходити від одновимірного часового ряду до латентного простору довільної розмірності для задоволення вимог інших компонентів моделі. Другий компонент – модель-декодувальник – подібний до першого, але замість перетворення одновимірних даних у високовимірні, він ставить у відповідність точкам латентного простору одновимірні точки з простору, у якому існують елементи часового ряду. Третій компонент моделі відповідає за визначення параметрів для системи нейронних звичайних диференціальних рівнянь, що має таку саму розмірність, як і латентний простір. Для системи з отриманими параметрами розв'язується задача Коші, початкові умови для якої отримуються моделлю-кодувальником. Час інтегрування задається залежно від бажаної довжини передбачення, а відстань у часі між сусідніми точками траєкторії – від бажаної точності або пропорційно частоті дискретизації модельованого часового ряду.

На основі нової моделі побудовано систему для синтезу мовлення з його визначних характеристик. Для цього реалізовано новий програмний продукт, що обчислює оцінки фундаментальної частоти, спектральних характеристик, та гучності для потокового сигнала.

Алгоритм оцінки фундаментальної частоти засновано на методі, що полягає у пошуку у сигналі послідовностей з чотирьох типів подій: позитивних перетинів нуля, негативних перетинів нуля, позитивних піків, та негативних піків. Значення інтервалів між подіями однакових типів усереднюються і на основі цього середнього значення обчислюється оцінка частоти. Додатково до самої оцінки також обчислюється її "впевненість" в межах від 0 до 1, яка чисельно виражає відхилення кожного зі знайдених інтервалів від їх середнього значення. Реалізація алгоритму враховує особливості предметної області, тому він виконує децимацію сигнала перед тим, як починати пошук описаних вище подій. Це дозволяє зменшити загальну кількість обчислень та пришвидшити роботу алгоритма. Для того, щоб уникнути втрати точності після децимації, при знаходженні можливих піків та перетинів нуля використовується відповідно квадратична та лінійна інтерполяція.

Алгоритм оцінки спектральних характеристик засновано на методі, який було адаптовано для застосування на потоковому аудіо. Ідея алгоритму полягає у тому, щоб виділяти для обчислення спектра сегмент сигнала, що містить у собі три періоди поточної фундаментальної частоти, та застосувати до нього серію згладжуючих перетворень. Це дозволяє більш точно оцінити стан сигнала в околі поточного семпла та уникнути небажаних варіацій, що залежать від часу.

Отримані визначні характеристики використовуються для отримання набору параметрів системи диференціальних рівнянь особливої форми за допомогою нейронної мережі. Обрання форми цієї системи було виконано з урахуванням особливостей предметної області. По-перше, модель має лінійний компонент, що дозволяє їй мати базові типи поведінки та моделювати прості періодичні процеси. По-друге, в модель включено нелінійний компонент, що дозволяє їй апроксимувати більш складну динаміку.

Наукова новизна отриманих результатів полягає у синтезуванні нової математичної моделі, яка складається з автономної системи звичайних диференціальних рівнянь, закону керування для неї, а також пари моделей для переходу від часової області до фазового простору побудованої моделі та назад. Структу-

ра моделі, на відміну від вже існуючих альтернативних підходів до синтезу мовлення, вперше дозволяє застосування традиційних методів аналізу звичайних диференціальних рівнянь для дослідження характеристик голосових сигналів у високорозмірному просторі. Завдяки своїй структурі, побудована модель є першою, що дозволяє синтезувати часові ряди з довільною точністю та частотою дискретизації, що значно спрощує проведення експериментів, які передбачають взаємодію з існуючими системами роботи з голосовими сигналами.

Під час розробки базової моделі, використовувалась нова узагальнена степенева активаційна функція, яка раніше не застосовувалася для вирішення подібних задач. Для того, щоб обґрунтувати використання такої функції, необхідно було суттєво розширити математичний апарат, чому і було присвячено значну частину другого розділу. Зокрема, було доведено теореми, які гарантують обмеженість рішень відповідних систем диференціальних рівнянь, наведено прості критерії існування інваріантних множин у цих системах, а також вказано можливі підходи до встановлення хаотичності отриманих розв'язків.

Ретельне доведення існування хаосу базувалося на побудові одновимірного дискретного відображення, що залежить від параметрів нової функції. Біфуркаційний аналіз цього відображення показав, що при деяких значеннях параметрів періодичні рішення системи диференціальних рівнянь, що описують динаміку мовлення, підкоряються процесу універсальності Фейгенбаума, наявність якого і призводила до появи хаотичних режимів.

Крім того, були проведені дослідження, що показали переваги введених степеневих активаційних функцій порівняно з такими відомими функціями як гіперболічний тангенс, сигмоїда та ReLU.

Практичне значення отриманих результатів полягає у розробленні програмного забезпечення, що реалізує нову модель (включно із її набором параметрів) та супутні алгоритми для виділення визначних характеристик потокового голосового сигналу для його параметричного синтезу. Виділені характеристики можна змінювати для керування сигналом, що генерується моделлю, або використовувати їх для інших прикладних задач.

Враховуючи природу моделі та можливість керування нею за допомогою легко інтерпретованих характеристик, легко бачити як розроблений програмний продукт можна використовувати як для подальших досліджень, так і на практиці. Наприклад, можна провести порівняльне дослідження різних голосо-

вих сигналів, що генеруються моделлю, та спробувати знайти зв'язок між динамічними характеристиками латентних траєкторій і типами звуків, що моделюються, провести біфуркаційних аналіз їх латентних траєкторій, тощо. До можливих практичних застосувань можна віднести використання автокодувальника, що було розроблено як частину моделі. Оскільки кодувальник гарантовано буде поміщати сегменти мовлення на атрактори (або близько до них) в латентному просторі, а декодувальник інтерпретує латентні траєкторії в аудіо сигнал, можна розробляти нові моделі або алгоритми, що будуть обробляти сигнали в латентному просторі (наприклад, визначати фундаментальну частоту або видаляти шум), що може виявитись більш ефективним, ніж робота з одновимірними часовими рядами.

У *вступі* обгрунтовано актуальність теми дослідження, наведено опис предмета та об'єкта досліджень, зазначено зв'язок дисертаційної роботи з науковими програмами, темами, підкреслена наукова новизна, мета дослідження, практичне значення, особистий внесок здобувача та інформацію щодо апробації отриманих результатів.

Перший розділ "Огляд предметної області та попередніх досліджень" присвячено висвітленню поняття динамічних систем, наведенню відомостей про методи їх моделювання, опису та аналізу класичних та нейромережевих підходів до побудови математичних аналогів реальних динамічних процесів. Також наведено огляд базових понять та методів обробки аналогових та дискретних сигналів у контексті роботи зі звуком, розглянуто голосовий сигнал як об'єкт дослідження, викладено загальні теоретичні відомості про його аналіз та синтез, інформацію про його структуру та головні характеристики, розглянуто та обґрунтовано моделювання голосового сигналу як динамічного процесу.

У *другому розділі* "Нова модель для розв'язання задачі керування системою ЗДР у латентному просторі" пропонується нова модель для апроксимації та прогнозування одновимірних часових рядів шляхом побудови адаптивного контролю для нейронної системи звичайних диференціальних рівнянь у високовимірному (латентному) просторі та побудові бієкції між точками латентних траєкторій та сегментами часового ряду. Нову модель випробувано для апроксимації та прогнозування простих періодичних сигналів і досліджено можливі проблеми, що можуть виникнути при її тренуванні та використанні. Також висвітлено проблему вибору архітектури та активаційної функції при побудо-

ві нових нейромережевих моделей і проведено порівняльний аналіз моделей з різними архітектурами та активаційними функціями, а також досліджено їх динамічні властивості.

У *третьому розділі* "Моделювання голосового сигналу як динамічного процесу в реальному часі" обґрунтовано вибір конкретних методів виділення головних характеристик голосового сигналу та безпосередньо описано реалізацію цих методів для практичного застосування. Також надається постановка задачі синтезу голосу та описано розв'язання як її, так і супутніх проблем.

У *загальних висновках* підсумовуються результати дисертаційної роботи та наводяться її основні здобутки.

Ключові слова: штучна нейронна мережа, степенева активаційна функція, система звичайних диференціальних рівнянь, динамічні системи, якісний аналіз динамічних систем, хаотичні процеси, аналіз часових рядів, методи вкладення часових рядів, апроксимація, оцінка фундаментальної частоти, перетворення Фур'є, аналіз спектральної оболонки, кепстральний аналіз, побудова оптимального контролю, гіпернейронні мережі.

Summary

Koshel Ye. V. Models and methods of speech synthesis based on neural systems of ordinary differential equations — Manuscript.

Thesis for the Doctor degree of Physical and Mathematical Sciences in Speciality 113 – Applied mathematics. — Oles Honchar Dnipro National Universityof Ministry of Education and Science of Ukraine, Dnipro, 2025.

The dissertation is devoted to the study of modern methods of speech synthesis based on the latest approaches to building neural network models. First of all, a new class of neural network models is considered – neural ordinary differential equations, which allows modeling dynamic processes in the form of systems of differential equations, the right-hand sides of which are defined as a neural network. The main advantage of this class of models in the context of the chosen subject area is that it allows using the well-known tools to study the qualitative characteristics of the constructed models.

During the work on the dissertation, a new model was developed based on neural ordinary differential equations, which allows approximating dynamic processes given by one-dimensional time series, as well as adapting to qualitative changes in these processes to preserve the quality of the approximation.

The developed model consists of several components that perform various tasks within its framework. The first such component is an encoder model that converts segments of the time series to points in the embedding space. This component allows to go from one-dimensional time series to a latent space of arbitrary dimension to satisfy the requirements of other components of the model. The second component – the decoder model – is similar to the first one, but instead of transforming one-dimensional data into high-dimensional data, it maps points of the latent space to one-dimensional points from the space in which the elements of the time series exist. The third component of the model is responsible for inferring the parameters for a system of neural ordinary differential equations, which has the same dimension as the latent space. For the system with the obtained parameters, the Cauchy problem is solved, the initial conditions for which are obtained by the encoder model. The integration time is set depending on the desired prediction length, and the distance in time between neighboring points of the trajectory – on the desired accuracy or proportional to the sampling frequency of the modeled time series.

Based on the new model, a system for speech synthesis from its distinctive

characteristics is built. For this purpose, a new software product has been implemented that calculates estimates of the fundamental frequency, spectral characteristics, and loudness for a streaming signal.

The fundamental frequency estimation algorithm is based on the DIO method, which consists in searching for sequences of four types of events in the signal: positive zero crossings, negative zero crossings, positive peaks, and negative peaks. The values of the intervals between events of the same types are averaged and the frequency estimate is calculated based on this average value. In addition to the estimate itself, its "confidence" is also calculated in the range from 0 to 1, which numerically expresses the deviation of each of the found intervals from their average value. The implementation of the algorithm takes into account the specifics of the subject area, so it performs signal decimation before starting the search for the events described above. This allows you to reduce the total number of calculations and speed up the algorithm. In order to avoid loss of accuracy after decimation, quadratic and linear interpolation are used when finding possible peaks and zero crossings respectively.

The algorithm for estimating spectral characteristics is based on the CheapTrick method, which was adapted for use in streaming audio. The idea of the algorithm is to allocate a signal segment containing three periods of the current fundamental frequency for spectrum calculation and apply a series of smoothing transformations to it. This allows for a more accurate assessment of the signal state in the vicinity of the current sample and to avoid unwanted time-dependent variations.

The obtained characteristic characteristics are used to obtain a set of parameters of a system of differential equations of a special form using a neural network. The choice of the form of this system was made taking into account the specifics of the subject area. First, the model has a linear component, which allows it to have basic types of behavior and simulate simple periodic processes. Secondly, the model includes a nonlinear component, which allows it to approximate more complex dynamics.

The scientific novelty of the obtained results lies in the synthesis of a new mathematical model, which consists of an autonomous system of ordinary differential equations, a control law for it, as well as a pair of models for the transition from the time domain to the phase space of the constructed model and back. The structure of the model, in contrast to existing alternative approaches to speech synthesis, for

the first time allows the use of traditional methods of analysis of ordinary differential equations to study the characteristics of voice signals in high-dimensional space. Due to its structure, the constructed model is the first to allow the synthesis of time series with arbitrary accuracy and sampling frequency, which greatly simplifies the conduct of experiments that involve interaction with existing systems for working with voice signals.

During the development of the basic model, a new generalized power activation function was used, which had not previously been applied in similar problems. In order to justify the use of such a function, it was necessary to significantly expand the mathematical apparatus, which is what a significant part of the second section was devoted to. In particular, theorems were proved that guarantee the boundedness of solutions to the corresponding systems of differential equations, simple criteria for the existence of invariant sets in these systems were given, and possible approaches to establishing the chaotic nature of the obtained solutions were also indicated.

A thorough proof of the existence of chaos was based on the construction of a one-dimensional discrete mapping that depends on the parameters of the new function. Bifurcation analysis of this mapping showed that for some values of the parameters, periodic solutions of the system of differential equations describing the dynamics of speech obey the Feigenbaum universality process, the presence of which led to the appearance of chaotic modes.

In addition, studies have been conducted that have shown the advantages of the introduced power-law activation functions compared to such well-known functions as hyperbolic tangent, sigmoid, and ReLU.

The practical significance of the results obtained lies in the development of software that implements the new model (including its set of parameters) and the accompanying algorithms for extracting the salient features of a streaming voice signal for its parametric synthesis. The extracted features can be modified to control the signal generated by the model or used for other applied tasks.

Given the nature of the model and the possibility of controlling it using easily interpreted features, it is easy to see how the developed software product can be used both for further research and in practice. For example, one can conduct a comparative study of different voice signals generated by the model and try to find a connection between the dynamic characteristics of latent trajectories and the types of sounds being modeled, conduct a bifurcation analysis of their latent

trajectories, etc. Possible practical applications include the use of an autoencoder that was developed as part of the model. Since the encoder is guaranteed to place speech segments on attractors (or close to them) in the latent space, and the decoder interprets the latent trajectories into an audio signal, it is possible to develop new models or algorithms that will process signals in the latent space (for example, determine the fundamental frequency or remove noise), which may be more efficient than working with one-dimensional time series.

The *introduction* justifies the relevance of the research topic, describes the subject and object of research, indicates the connection of the dissertation work with scientific programs, topics, emphasizes scientific novelty, the purpose of the research, practical significance, personal contribution of the applicant, and information on the approval of the results obtained.

The first section "Overview of the subject area and previous research" is devoted to highlighting the concept of dynamic systems, providing information on methods for their modeling, description and analysis of classical and neural network approaches to constructing mathematical analogues of real dynamic processes. Also, an overview of basic concepts and methods for processing analog and discrete signals in the context of working with sound is provided, the voice signal as an object of study is considered, general theoretical information about its analysis and synthesis, information about its structure and main characteristics is presented, the feasibility of modeling the voice signal as a dynamic process is considered and substantiated.

In the *second section* "A New Model for Solving the Problem of Controlling a LDC System in Latent Space", a new model is proposed for approximating and predicting one-dimensional time series by constructing adaptive control for a neural system of ordinary differential equations in high-dimensional (latent) space and constructing a bijection between points of latent trajectories and segments of the time series. The new model is tested for approximating and predicting simple periodic signals and possible problems that may arise during its training and use are investigated. The problem of choosing architecture and activation functions when building new neural network models is also highlighted and a comparative analysis of models with different architectures and activation functions is carried out, as well as their dynamic properties are investigated.

In the *third section* "Modeling a voice signal as a dynamic process in real time", the choice of specific methods for isolating the main characteristics of a voice signal

is justified and the implementation of these methods for practical application is directly described. The problem of voice synthesis is also stated and the solution of both it and related problems is described.

In the *general conclusions*, the results of the dissertation work are summarized and its main achievements are given.

Keywords: artificial neural network, power activation function, system of ordinary differential equations, dynamical systems, qualitative analysis of dynamical systems, chaotic processes, time series analysis, approximation, time series nesting methods, fundamental frequency estimation, Fourier transform, spectral envelope analysis, cepstral analysis, optimal control construction, hyperneural networks

Список публікацій здобувача

*Статті у наукових фахових виданнях України, які входять до міжнародних наукометричних баз даних **Scopus**:*

1. Belozyorov V. Ye., Dantsev D. V., Koshel Y. V. On the Equivalence of Real Dynamic Process and Its Models. – Journal of Optimization, Differential Equations and Their Applications. DOI 10.15421/142109, ISSN (print) 2617–0108, ISSN (on-line) 2663–6824. - V. **29**. - №2.- December 2021. - P. 76-91. *(Особистий внесок: розробка нового метода для обчислення значень локального якобіану для точок, реконструйованого з часового ряду фазового простору, атрактора, що дозволило застосувати алгоритм для обчислення показників Ляпунова для системи, аналітична модель якої невідома; Білозьоров В.Є: проектування та дослідження загальної системи квадратичних диференціальних рівнянь, що описують поведінку будь-якого динамічної процесу, а також обчислення параметрів цих рівнянь, які визначаються заданим часовим рядом; Данцев Д.В: дослідження всіх прикладів, які продемонстровані в статті).*

2. Belozyorov V. Ye., Koshel Y. V. Univariate time series analysis with hyperneural ODE. – Journal of Optimization, Differential Equations and Their Applications. DOI 10.15421/142311, ISSN (on-line) 2663–6824. - V. **31**. - №2.- December 2023. - P. 50-66. *(Особистий внесок: запропоновано нову нейромережеву модель на базі звичайних диференціальних рівнянь – hNODE, що дозволяє моделювати нестаціонарні часові ряди, а також проведено аналіз впливу активаційної функції, розмірності, та кроку інтегрування на якість навчання та роботи моделі; Білозьоров В.Є: запровадження поняття непарної активаційної функції, що використовується здобувачем у всіх дослідженнях цієї статті).*

3. Belozyorov V. Ye., Koshel Y. V. On systems of neural ODEs with generalized power activation functions. - Journal of Optimization, Differential Equations and Their Applications. DOI 10.15421/141901, ISSN (print) 2617–0108, ISSN (on-line) 2663–6824. - V. **32**. - №2.- December 2024. - P. 56 – 91. *(Особистий внесок: запропоновано та досліджено властивості нового типу гіпер нейронних ЗДР, що включають в себе лінійні та нелінійні компонен-*

ти, а також проведено його порівняльний аналіз з іншими можливими типами моделей, сформульовано та доведено Леми 2.1, 2.2 про існування положень рівноваги в системі (2.28)*, а також доведено Лему 2.3, яка присвячена біфуркаційному аналізу відображення* (2.54)*; Білозьоров В.Є.: формулювання теорем 2.4, 2.6, 2.7, 2.8 та 2.10, розробка прикладів 3D систем, які демонструють хаотичну поведінку).*

4. Belozyorov V. Ye., Koshel Ye. V., Zaytsev V.G. The possibility of the robust chaos emergence in the Lorenz-like nonautonomous system – Journal of Optimization, Differential Equations and Their Applications. DOI 10.15421/141902, ISSN (print) 2617–0108, ISSN (on-line) 2663–6824. - V. **27**. - №2.- December 2019. - P. 44-59. *(Особистий внесок: біфуркаційний аналіз досліджуваної системи та визначення параметрів, за яких досягається придушення хаосу; Білозьоров В.Є.: проведення рекурентного аналізу отриманих систем; Зайцев В.Г.: проведення чисельних експериментів та побудова рекурентних діаграм).*

Статті у наукових фахових виданнях України категорії Б:

1. Koshel Y. V. Adaptation of the world framework for frame-by-frame real-time speech analysis – System technologies. DOI 10.34185/1562-9945-5-148-2023-03, ISSN (print) 1562-9945, ISSN (on-line) 2707-7977. – V. **5** – №148 – 2023 – P. 21–36.

2. Koshel Y. V. Neural network-assisted continuous embedding of univariate data streams for time series analysis – System technologies. DOI10.34185/1562-9945-2-151-2024-08, ISSN (print) 1562-9945, ISSN (on-line) 2707-7977. – V. **2** – № 151 – 2024 – P. 92–101.

Тези наукових доповідей:

1. Koshel E. Local Jacobian estimation for delay embedded time series data. Тези доповіді на 18-й міжнародній науково-практичній конференції "Математичне та програмне забезпечення інтелектуальних систем" (MPZIS-2020). – 18 – 20 листопада 2020 р. Дніпро, с. 152 – 153.

2. Koshel E. Lyapunov spectrum estimation for delay embedded time series. – Тези доповіді на 19-й міжнародній науково-практичній конференції "Ма-

тематичне та програмне забезпечення інтелектуальних систем" (MPZIS-2021). – 17 – 19 листопада 2021 р. Дніпро, с. 110–111.

3. Koshel Y. V., Belozyorov V. Ye. Modelling Dynamical Processes Described By Time Series Data With Neural Differential Equations. – Тези доповіді на 20-й міжнародній науково-практичній конференції "Математичне та програмне забезпечення інтелектуальних систем" (MPZIS-2022). – 23–25 листопада 2022 р. Дніпро, с. 111–112.

4. Koshel Eu. Hyper neural ODE for univariate time series analysis, – Тези доповіді на 21-й міжнародній науково-практичній конференції "Математичне та програмне забезпечення інтелектуальних систем" (MPZIS-2023). – 22–24 листопада 2023, Дніпро, с. 21–22.

5. Koshel Y. V, Belozyorov V. Ye. Generation of stable systems of ordinary differential equations for univariate time series modeling. - Тези доповіді на XXVII МІЖНАРОДНІЙ КОНФЕРЕНЦІЇ З АВТОМАТИЧНОГО КЕРУВАННЯ «АВТОМАТИКА 2024» – 20 – 22 листопада 2024 р., Дніпро, С. 36 – 37.

6. Belozyorov V. Ye, Koshel Y. V. On use of power activation functions with a degree less than 1 in neural networks. - Тези доповіді на 22-й міжнародній науково-практичній конференції "Математичне та програмне забезпечення інтелектуальних систем" (MPZIS-2024). – 20—22 листопада 2024 р., Дніпро, С. 7 – 9.

7. Koshel Eu. Improvement for peaks and zero-crossings detection for time-domain pitch estimation algorithms – Proceedings of 1st international scientific and practical conference "CURRENT CHALLENGES OF SCIENCE AND EDUCATION", – September 18–20 2023, Berlin, – P. 116–118.

Зміст

Список скорочень **18**

Вступ **20**

1 Огляд предметної області та попередніх досліджень **28**

1.1 Моделювання динамічних систем 28

1.1.1 Класичні методи 34

1.1.2 Нейромережевий підхід 40

1.2 Обробка сигналів 46

1.2.1 Аналогові сигнали 49

1.2.2 Дискретні сигнали 53

1.2.3 Частотний аналіз 55

1.2.4 Спектрограма 59

1.2.5 Фільтрування 61

1.2.6 Кепстральний аналіз 63

1.3 Голосовий сигнал як об'єкт дослідження 64

1.4 Висновки 68

2 Нова модель для розв'язання задачі керування системою ЗДР у латентному просторі **70**

2.1 Вибір активаційної функції для нейромережевого моделювання часових рядів 74

2.2 Прогнозування часових рядів без явної параметризації 81

2.3 Прогнозування часових рядів з явною параметризацією шляхом включення параметрів в контекст 84

2.4 Порівняння впливу різних активаційних функцій на якість прогнозування 85

2.5 Можливості для виникнення некоректної поведінки під час тренування моделі hNODE . . . 87
2.6 Обґрунтування вибору структури моделі та застосованих в ній активаційних функцій . . . 88
2.6.1 Обмеженість розв'язків системи нейронних ЗДР . . . 89
2.6.2 Строге доведення існування хаосу в системі (2.28) . . . 102
2.6.3 Стосовно системи (2.28) без лінійної частини . . . 108
2.6.4 Порівняння нейронних мереж з різними активаційними функціями . . . 111
2.7 Висновки . . . 125

3 Моделювання голосового сигналу як динамічного процесу 126
3.1 Реалізація оцінювання фундаментальної частоти для потокового аудіо сигналу . . . 126
3.2 Реалізація оцінювання спектральних характеристик голосу для потокового аудіо сигналу . . . 132
3.3 Оцінка якості розроблених методів оцінювання характеристик мовлення . . . 136
3.4 Вибір та реалізація методу переходу від одновимірного часового ряду до багатовимірного латентного простору та назад . . . 139
3.5 hNODE модель для параметричного генерування голосового сигналу . . . 149
3.5.1 Розбиття вхідного сигналу на сегменти та попередня обробка сегментів . . . 149
3.5.2 Вибір архітектури кодувальника та декодувальника . . . 155
3.5.3 Вибір архітектури для вкладеної NODE та моделі для генерування її параметрів . . . 157
3.5.4 Функція втрат для тренування моделі та підготовка навчальних даних . . . 161
3.6 Перевірка роботи моделі на тестовому сигналі . . . 166
3.7 Висновки . . . 169

Загальні висновки 170

Список використаних джерел 172

Додаток А **184**

Список скорочень

- CDIO – реалізація алгоритма DIO як частини фреймворка WORLD мовою C++.
- CheapTrick – алгоритм для оцінювання спектральної оболонки сигналу.
- DFT – дискретне перетворення Фур'є (Discrete Fourier Transform).
- DIO – алгоритм для оцінки фундаментальної частоти (Distributed Inline Filtering with Overlap).
- FFT – швидке перетворення Фур'є (Fast Fourier Transform).
- FIR – фільтр зі скінченними імпульсними характеристиками (Finite Impulse Response filter).
- hNODE – гіпер нейронні системи звичайних диференціальних рівнянь (Hyper Neural Ordinary Differential Equations), тобто NODE, параметри яких залежать від вхідних даних.
- IDFT – обернене дискретне перетворення Фур'є (Inverse Discrete Fourier Transform).
- IIR – фільтр з нескінченними імпульсними характеристиками (Infinite Impulse Response filter).
- JCheapTrick – реалізація алгоритму CheapTrick мовою Julia.
- JDIO – реалізація алгоритму DIO мовою Julia.
- MFCC – мелчастотні кепстральні коефіцієнти (Mel-Frequency Cepstral Coefficients).

- MNIST – база даних зображень рукописних цифр та їх лейблів (Modified National Institute of Standards and Technology database).
- NN – нейронна мережа (Neural Network).
- NODE – нейронні системи звичайних диференціальних рівнянь (Neural Ordinary Differential Equations).
- PECUZAL – уніфікований та автоматизований підхід до реконструкції атрактора названий на честь дослідників, чия робота лягла в його основу – Луї Пекори та Лукаса Узала (Pecora-Uzal).
- PINN – фізично поінформована нейромережа (Physics-Informed Neural Network).
- PSD – спектральна густина сигналу (Power Spectral Density).
- SI – ідентифікація мовця (Speaker Identification).
- STFT – віконне перетворення Фур'є (Short-Time Fourier Transform).
- STT – голосовий набір тексту (Speech-To-Text).
- TTS – автоматичне озвучення тексту (Text-To-Speech).
- VC – заміна мовця (Voice Conversion).
- WT – вейвлетне перетворення (Wavelet Transform).
- YDIO – референсна реалізація алгоритма DIO інженерами компанії Yamaha.
- АІС – адаптивна ідентифікація систем.
- ЗДР – звичайні диференціальні рівняння.
- ІС – ідентифікація систем.
- ПЗ – програмне забезпечення.

Вступ

Актуальність теми. Аналіз та синтез мовлення завжди були популярними темами для багатьох дослідників. Підходи та методи роботи із голосовими сигналами, що були розроблені протягом останніх десятиліть, зараз застосовуються для розв'язання найрізноманітніших задач, таких як озвучення тексту (text-to-speech, TTS) [1]—[6], ідентифікація мовця (speaker identification, SI) [7]—[12], заміна мовця (voice conversion, VC) [13]—[15], голосовий набір тексту (speech-to-text, STT) [16]—[19], та інші. На основі цих розробок створюються системи для голосового контролю над програмним забезпеченням (ПЗ) комп'ютерів, а також дозволяють цьому ПЗ комунікувати з користувачами за допомогою синтезованого голосу. Подібні системи додають новий інтерфейс, який дозволяє безконтактну взаємодію користувача із пристроєм, що значно розширює коло сценаріїв для взаємодії людини й машини, а також робить комп'ютери більш доступними для людей з обмеженими можливостями.

Вагомий внесок у дослідження аналізу та синтезу мовлення, а також у стрімкий розвиток інструментів для роботи з ним зробили Гомер Дадлі, Франклін С. Купер, Алвін Ліберман, Джон Л. Келлі, Луї Ґерстман, Фумітада Ітакура, Шуцо Сайто, Бішну С. Атал, Манфред Р. Шредер, Денніс Клатт, та багато інших.

На момент написання цієї роботи, найбільш потужним та популярним інструментом для моделювання є штучні нейронні мережі [20]—[22], і саме вони лежать в основі більшості систем аналізу та синтезу голосових сигналів. Головна перевага нейронних мереж над іншими методами моделювання полягає у їхньому підході до представлення даних та операцій над ними. Будь-який об'єкт, що моделюється нейронною мережею, може і має бути представлений у вигляді вектора, компоненти якого відповідають окремим характеристикам об'єкта та чисельно описують їх. Операції над векторним представленням об'єктів у переважній більшості випадків моделюються як їх множення на матрицю, додавання до них вектора зсуву, та застосування до результату деякої нелінійної

“функції активації”:

$$y = F(x) = \sigma(Wx + b), \tag{0.1}$$

тут W – матриця коефіцієнтів, b – вектор зсуву, а σ – функція активації.

Такий підхід до представлення даних та операції дозволяє нейронним мережам легко масштабуватися як горизонтально, так і вертикально. Горизонтальне масштабування досягається шляхом збільшення розмірності матриці W та вектора b, що дозволяє моделі обирати більше лінійних комбінацій компонентів векторизованого об’єкта x та потенційно “вивчати” більше зв’язків між ними [23]. Вертикальне масштабування реалізується як послідовне застосування декількох нейромережевих моделей до вхідного вектора:

$$\begin{aligned} y_1 &= F_1(x) = \sigma_1(W_1 x + b_1) \\ y_2 &= F_2(x) = \sigma_2(W_2 y_1 + b_2) \\ &\dots \\ y_n &= F_n(x) = \sigma_n(W_n y_{n-1} + b_n), \end{aligned} \tag{0.2}$$

де n – глибина нейронної мережі, y_i – проміжний результат перетворення вхідного вектора x, та $y = y_n$ – фінальний результат перетворень. Проте конкретний тип масштабування може по-різному впливати на різні типи задач [24], [25]. Тривіальність масштабування нейромережевих моделей, здатність процесорів швидко знаходити добуток матриць, а також новітні методи оптимізації функцій багатьох змінних привели до вибухового прогресу у галузі машинного навчання. Гнучкі та потужні нейронні мережі найрізноманітніших архітектур лягли в основу проєктів, які ще десяток років тому були мислимими лише як теми для науково-фантастичної літератури. Сьогодні програмне забезпечення на основі нейронних мереж може створювати музичні твори, писати картини, виконувати функції персонального асистента та правдоподібно підтримувати діалог.

Але у такого неймовірного прогресу є і зворотна сторона. Як згадувалося вище, потужність нейромережевої моделі прямо пропорційна її розміру, тобто найпростіший спосіб покращення якості роботи моделі - це її горизонтальне або вертикальне масштабування. Але чим більшою стає модель, тим більше обчислювальних ресурсів вона буде потребувати, а найгірше - нову масштабовану модель необхідно буде оптимізувати наново, оскільки розмірність параметрів

меншої моделі буде несумісною із розмірністю параметрів нової, більшої. Таким чином, кожна нова ітерація конкретної моделі буде потребувати перенавчання та збільшення обчислювальної потужності інфраструктури, на якій розгорнуто цю модель. Саме тому головним напрямом досліджень наразі є створення більш ефективних моделей та архітектур, компоненти яких можна краще адаптувати до області застосування та масштабувати їх незалежно одне від одного. Адаптація до предметних областей здебільшого полягає у структуризації моделі таким чином, щоб вона не могла порушувати інваріанти даної області. А задача розбиття моделі на компоненти є тотожною задачі декомпозиції проблеми, яку має вирішити модель, тобто для кожної підзадачі може розроблятися окрема спеціалізована модель.

Зв'язок роботи з науковими програмами, планами, темами. Робота виконана в рамках ініціативної науково-дослідної роботи "Детерміновані та стохастичні алгоритми комп'ютерного моделювання об'єктів та процесів різної природи" кафедри комп'ютерних технологій факультету прикладної математики та інформаційних технологій Дніпровського національного університету імені Олеся Гончара, номер державної реєстрації 0122U001467.

Мета і завдання роботи – дослідити сучасні методи та підходи до аналізу та синтезу мовлення за допомогою нейромережевих моделей, а також:

- Розробити нову модель на основі нейронних систем звичайних диференціальних рівнянь для синтезу мовлення.
- Забезпечити можливість використання моделі в різних середовищах без втрати попереднього навчання.
- Дослідити вплив архітектури та функцій активації на якісні характеристики побудованої моделі.
- Передбачити можливість використання класичних методів для дослідження властивостей моделі.
- Забезпечити максимальну прозорість та інтерпретовність нової моделі.

Об'єктом дослідження є звукові коливання, що породжуються голосовим апаратом людини.

Предметом дослідження є моделювання нестаціонарних часових рядів, отриманих шляхом аналогово-цифрового перетворення голосового сигналу, як динамічних систем за допомогою нейромережевих моделей.

Методи дослідження, використані для досягнення мети роботи, включають аналіз, порівняння, моделювання, та експеримент. Застосування аналізу полягало у розділенні досліджуваного сигналу на головні компоненти як для їх самостійного вивчення, так і для вивчення їх взаємозв'язків. Метод порівняння було використано для визначення спільних та відмінних рис між системою, утвореною гортанню та голосовим трактом людини, та іншими системами породження звуків. За результатами аналізу та порівняння було застосовано метод моделювання для створення математичної моделі, у визначення якої входять нейромережеві компоненти, для створення цифрового аналога голосової системи людини. З використанням створеної моделі було застосовано експериментальний метод дослідження для більш детального розгляду впливу числових характеристик голосу на динамічну поведінку дискретизованих часових рядів.

Наукова новизна отриманих результатів полягає у синтезуванні нової математичної моделі, яка складається з автономної системи звичайних диференціальних рівнянь, закону керування для неї, а також пари моделей для переходу від часової області до фазового простору побудованої моделі та назад. Структура моделі, на відміну від вже існуючих альтернативних підходів до синтезу мовлення, вперше дозволяє застосування традиційних методів аналізу звичайних диференціальних рівнянь для дослідження характеристик голосових сигналів у високорозмірному просторі. Завдяки своїй структурі, побудована модель є першою, що дозволяє синтезувати часові ряди з довільною точністю та частотою дискретизації, що значно спрощує проведення експериментів, які передбачають взаємодію з існуючими системами роботи з голосовими сигналами.

Практичне значення отриманих результатів полягає у розробленні програмного забезпечення, що реалізує нову модель (включно із її набором параметрів) та супутні алгоритми для виділення визначних характеристик потокового голосового сигналу для його параметричного синтезу. Виділені характеристики можна змінювати для керування сигналом, що генерується моделлю, або використовувати їх для інших прикладних задач.

Враховуючи природу моделі та можливість керування нею за допомогою легко інтерпретованих характеристик, легко бачити як розроблений програм-

ний продукт можна використовувати як для подальших досліджень, так і на практиці. Наприклад, можна провести порівняльне дослідження різних голосових сигналів, що генеруються моделлю, та спробувати знайти зв'язок між динамічними характеристиками латентних траєкторій і типами звуків, що моделюються, провести біфуркаційних аналіз їх латентних траєкторій, тощо. До можливих практичних застосувань можна віднести використання автокодувальника, що було розроблено як частину моделі. Оскільки кодувальник гарантовано буде поміщати сегменти мовлення на атрактори (або близько до них) в латентному просторі, а декодувальник інтерпретує латентні траєкторії в аудіо сигнал, можна розробляти нові моделі або алгоритми, що будуть обробляти сигнали в латентному просторі (наприклад, визначати фундаментальну частоту або видаляти шум), що може виявитись більш ефективним, ніж робота з одновимірними часовими рядами.

Особистий внесок здобувача у роботах, що опубліковані у співавторстві, наступні. У статті [26] здобувач розробив новий метод для обчислення значень локального якобіану для точок, реконструйованого з часового ряду фазового простору, атрактора, що дозволило застосувати алгоритм для обчислення показників Ляпунова для системи, аналітична модель якої невідома. Другому співавтору Білозьорову В.Є. належить проектування та дослідження загальної системи квадратичних диференціальних рівнянь, що описують поведінку будь-якого динамічної процесу, а також обчислення параметрів цих рівнянь, які визначаються заданим часовим рядом. Третьому співавтору Данцеву Д.В. належать дослідження всіх прикладів, які продемонстровані в статті.

У статті [27] здобувачем запропоновано нову нейромережеву модель на базі звичайних диференціальних рівнянь – hNODE, що дозволяє моделювати нестаціонарні часові ряди, а також проведено аналіз впливу активаційної функції, розмірності, та кроку інтегрування на якість навчання та роботи моделі. Другому співавтору Білозьорову В.Є. належить запровадження поняття непарної активаційної функції, що використовується здобувачем у всіх подальших дослідженнях цієї статті.

У статті [28] здобувачем запропоновано та досліджено властивості нового типу гіпер нейронних ЗДР, що включають в себе лінійні та нелінійні компоненти, а також проведено його порівняльний аналіз з іншими можливими типами моделей. Було сформульовано та доведено Леми 2.1, 2.2 про існування положень

рівноваги в системі (2.28), а також доведено Лему 2.3, яка присвячена біфуркаційному аналізу відображення (2.54). Формулювання теорем 2.4, 2.6, 2.7, 2.8 та 2.10 було зроблено проф. Білозьоровим В.Є., а їх доведення було зроблено здобувачем після деяких зауважень Білозьорова В.Є. Ці зауваження полягали в тому, що перед доведення обмеженості рішень системи (2.28) потрібно встановити границі її інваріантних багатостатностей. Це завдання було вирішено здобувачем для системи (2.28) зі степеневими активаційними функціями. Другому співавтору Білозьорову В.Є. належить розробка прикладів 3D систем, які демонструють хаотичну поведінку.

Апробація матеріалів дисертації. За результатами дисертаційної роботи було взято участь у конференціях:

- 18-а міжнародна науково-практична конференція "Математичне та програмне забезпечення інтелектуальних систем" (MPZIS-2020).
- 19-а міжнародна науково-практична конференція "Математичне та програмне забезпечення інтелектуальних систем" (MPZIS-2021).
- 20-а міжнародна науково-практична конференція "Математичне та програмне забезпечення інтелектуальних систем" (MPZIS-2022).
- 21-а міжнародна науково-практична конференція "Математичне та програмне забезпечення інтелектуальних систем" (MPZIS-2023).
- 27-а міжнародна конференція з автоматичного керування "Автоматика 2024".
- 22-а міжнародна науково-практична конференція "Математичне та програмне забезпечення інтелектуальних систем" (MPZIS-2024).

Структура та обсяг дисертації наступні:

- *Анотація*, що наведена українською та англійською мовами, дає загальний огляд роботи та коротко викладає її вміст.
- *Зміст* наводить ієрархічний список структурних елементів дисертаційної роботи.

- *Список скорочень* містить список скорочень та абревіатур, що широко використовуються у тексті роботи.

- *Вступ* містить обґрунтування вибору теми дослідження, мету і завдання дослідження відповідно до предмета та об'єкта дослідження, зв'язок роботи з державними науковими програмами, методи дослідження, інформацію про наукову новизну отриманих результатів, особистий внесок здобувача, апробацію матеріалів дисертації, а також структуру та обсяг дисертації.

  У *вступі* обгрунтовано актуальність теми дослідження, наведено опис предмета та об'єкта досліджень, зазначено зв'язок дисертаційної роботи з науковими програмами, темами, підкреслена наукова новизна, мета дослідження, практичне значення, особистий внесок здобувача та інформацію щодо апробації отриманих результатів.

- *Перший розділ* "Огляд предметної області та попередніх досліджень" присвячено висвітленню поняття динамічних систем, наведенню відомостей про методи їх моделювання, опису та аналізу класичних та нейромережевих підходів до побудови математичних аналогів реальних динамічних процесів. Також наведено огляд базових понять та методів обробки аналогових та дискретних сигналів у контексті роботи зі звуком, розглянуто голосовий сигнал як об'єкт дослідження, викладено загальні теоретичні відомості про його аналіз та синтез, інформацію про його структуру та головні характеристики, розглянуто та обґрунтовано доцільність моделювання голосового сигналу як динамічного процесу.

- У *другому розділі* "Нова модель для розв'язання задачі керування системою ЗДР у латентному просторі" пропонується нова модель для апроксимації та прогнозування одновимірних часових рядів шляхом побудови адаптивного контролю для нейронної системи звичайних диференціальних рівнянь у високовимірному (латентному) просторі та побудові бієкції між точками латентних траєкторій та сегментами часового ряду. Нову модель випробувано для апроксимації та прогнозування простих періодичних сигналів і досліджено можливі проблеми, що можуть виникнути при її тренуванні та використанні. Також висвітлено проблему вибору архітектури та активаційних функцій при побудові нових нейромережевих моде-

лей і проведено порівняльний аналіз моделей з різними архітектурами та активаційними функціями, а також досліджено їх динамічні властивості

- У *третьому розділі* “Моделювання голосового сигналу як динамічного процесу в реальному часі” обґрунтовано вибір конкретних методів виділення головних характеристик голосового сигналу та безпосередньо описано реалізацію цих методів для практичного застосування. Також надається постановка задачі синтезу голосу та описано розв’язання як її, так і супутніх проблем.

- *Четвертий розділ* “Загальні висновки” підсумовує усі попередні розділи, наводить найбільш важливі наукові та практичні результати дисертації, вказує проблеми та задачі, для розв’язання яких можуть бути використані отримані результати, та перераховує можливі напрями продовження досліджень за тематикою роботи.

Загальний обсяг основного тексту роботи становить 186 сторінок. Робота містить 54 рисунки і 3 таблиці, список використаних джерел налічує 121 найменування.

Розділ 1

Огляд предметної області та попередніх досліджень

1.1 Моделювання динамічних систем

Ідентифікація природних систем, їх компонентів та законів розвитку є не тільки метою науки як виду людської діяльності, але і запо рукою виживання людей як біологічного виду. Система – це сутність, яка сприймається як єдиний об'єкт, що складається із взаємопов'язаних елементів. Елементи системи також можуть бути системами, а система, до якої вони належать, може бути підсистемою більшої системи.

З такого напіврекурсивного визначення випливає, що системами не можуть вважатися об'єкти, що не містять у собі жодних компонентів: порожня множина або точка в евклідовому просторі не є системами [29]; водночас системи, що містять у собі лише один об'єкт, можна вважати тотожними даному об'єкту, оскільки включення об'єкта у таку систему ніяк не вплине на його поведінку та не змінить кількість інформації, яку можна з нього отримати. Звідси маємо визначення нижньої границі кількості об'єктів у системі – для того, щоб вважатися системою, об'єкт має складатися як мінімум з двох компонентів. Визначення верхньої границі є набагато більш проблематичним, але ми можемо використати той факт, що об'єкти в системі мають бути певним чином взаємопов'язаними, щоб вважатися її компонентами, та спробувати сформувати правила, за якими з системи можна вилучати компоненти задля спрощення роботи з нею. Слушно припустити, що природною системою із найбільшою кількістю компонентів є

сам Всесвіт, але чи усі його компоненти є взаємопов'язаними? Дійсно, більшість відомих об'єктів у Всесвіті так чи інакше взаємодіють одне з одним через електромагнітні, гравітаційні, та інші фізичні поля. Але через астрономічні відстані між ними, більшість ефектів цих взаємодій майже неможливо виміряти. До того ж існують відстані, на яких деякі види взаємодії стають просто неможливими – наприклад, неможливо у видимому спектрі побачити об'єкти, що знаходяться далі за 13.7 млрд світлових років від Землі. З цього випливає, що системи можуть містити компоненти, вплив яких на загальну поведінку настільки неістотний, що ним можна знехтувати, а самі компоненти вилучити з системи. Мінімальна кількість об'єктів, які неможливо вилучити з системи істотно не вплинувши на її поведінку, і є верхньою границею кількості об'єктів у даній системі.

Іншою важливою характеристикою систем є їх здатність змінюватися з часом. Ця характеристика поділяє усі системи на статичні та динамічні. Переважна більшість природних систем є динамічними – стан їх компонентів змінюється з часом внаслідок їх взаємодії між собою та під впливом зовнішніх чинників. Статичні системи натомість майже не зустрічаються у природі, але саме вони часто є основою для інструментів та приладів, що створюються людьми. Прикладом такого інструменту є логарифмічна лінійка – вона складається з багатьох взаємопов'язаних компонентів, вилучення будь-якого з них значно вплине на її поведінку, і при цьому жоден з цих компонентів не здатен самостійно змінювати свій стан у часі. Однак логарифмічна лінійка значно відрізняється від інших фізичних систем, які зустрічаються у природі. Вона є інструментом, що кодує певні математичні відношення між величинами; логарифмічна лінійка – це математична модель.

Математична модель – це інструмент досліджень, зміст якого полягає у представленні процесів або співвідношень у вигляді математичних рівнянь, нерівностей, обмежень, або їх систем. Метою математичного моделювання є побудова *аналога* модельованої системи для пошуку її невідомих параметрів або передбачення майбутніх значень. Логарифмічна лінійка у прикладі вище – це статична математична модель, яка, залежно від її конкретної реалізації, моделює операції множення, ділення, піднесення до степеня, добування кореня тощо. Логарифмічна лінійка та подібні їй аналогові обчислювальні інструменти демонструють важливий аспект як моделювання, так і практичного застосу-

вання побудованих моделей – вибір правильної перспективи. Якщо розглянути процес застосування математичних моделей абстрактно, то легко побачити, що він полягає у пошуці перетину системи, яка моделюється, та даних, отриманих з моделі. У випадку з логарифмічною лінійкою, модельованою системою є числа та операції над ними, а дані з моделі отримуються шляхом пересуванням повзунків лінійкою та зчитування чисел із нанесених на неї шкал. Але даний приклад не повністю висвітлює тезу про співставлення штучної моделі та природної системи, оскільки сама арифметика вже є своєрідною моделлю навколишнього світу. Більш коректно у цій ситуації було б навести як приклад інструменти для вимірювання відстані, такі як звичайна лінійка, секстант, лазерний далекомір тощо. Рівняння, які фізично реалізують подібні інструменти, описують математичну модель поняття відстані в Евклідовому просторі, кутової відстані, взаємодії швидкості руху елементарних частинок та часу. І хоча подібні фізичні моделі є потужним інструментом досліджень, побудова фізичного аналога модельованої системи часто не є можливим. У такому разі дослідникам залишається вдаватися до суто абстрактних, теоретичних методів аналізу, які полягають у математичних перетвореннях компонентів побудованої моделі, поданої у вигляді рівнянь.

Не варто думати, що абстрактний підхід до моделювання менш ефективний за практичний і навпаки. Сторіччя наукового прогресу укорінили математику як безальтернативну мову описання законів природи [30]. Системи аксіом, правила перетворень математичних об'єктів та їх взаємодії між собою у сучасній математиці є нічим іншим як абстрактною моделлю процесів та взаємозв'язків між об'єктами в реальному світі. Неминучим наслідком високого рівня абстракції у математиці є викриття спільних рис між процесами та об'єктами, які не мають між собою нічого спільного з емпіричної точки зору. Наочним прикладом подібних зв'язків є формули для обчислення сили гравітації та сили взаємодії між двома точковими зарядами:

$$F = G\frac{m_1 m_2}{r^2}, \tag{1.1}$$

$$F = k\frac{q_1 q_2}{r^2}. \tag{1.2}$$

Не зважаючи на те, що закон всесвітнього тяжіння (1.1) описує взаємодії, що відбуваються в гравітаційному полі, а закон Кулона (1.2) – взаємодії стати-

чних зарядів в електростатичному полі, абстрактне математичне представлення обох цих взаємодій ідентичне. Можна сказати, що обидва ці закони побудовані на одній і тій самій моделі, яка відрізняється лише параметрами та інтерпретацією. Параметрами у першому випадку є гравітаційна стала та маси двох тіл, у другому – електростатична стала та значення зарядів, і обидва містять квадрат відстані як елемент оберненої пропорційності. Отже, навіть проста статична модель здатна точно описати два феномени з різних галузей фізики та пояснити зв'язки між ними.

Потужність та гнучкість математичних підходів до аналізу моделей природних явищ дозволяє проведення суто теоретичних досліджень для отримання нових результатів, які потім можна підтвердити або спростувати експериментально. За такого підходу експерименти є набагато більш економічними та направленими на конкретний результат, оскільки їх вхідні та цільові значення є заздалегідь прорахованими, і саме це дозволило фізиці поділитися на дві гармонійні половини – теоретичну та експериментальну.

З наведених вище прикладів видно, що статичні математичні моделі є потужним інструментом для пізнання світу. Але існує ще один клас моделей, які здатні описувати не тільки поточний стан системи, але і закон його зміни з часом. Стан у таких системах зазвичай описується алгебраїчно у вигляді вектора чисел, а послідовність станів, яких набуває система з плином часу, називають її *траєкторією*. Детальним вивченням цього класу моделей займається теорія динамічних систем.

Теорія динамічних систем – це розділ математики, у якому досліджують поведінку динамічних систем з часом. Динамічна система – це математична модель, яка складається з множини станів системи – точок деякого абстрактного простору ξ, який називають фазовим простором, – та закону, що визначає еволюцію станів за час t і задається відображеннями Φ_t простору ξ самого в себе. Простір та відображення завжди задовольняють певні умови. Найчастіше фазовим простором є n-вимірний (евклідів) простір, а відображення Φ_t є неперервними чи диференційовними та утворюють групу відносно t, тобто $\Phi_{t_3} = \Phi_{t_2}(\Phi_{t_1})$ [31]. В рамках теорії динамічних систем зазвичай розглядаються детерміновані відображення Φ_t, тобто такі, визначення яких не містить у собі стохастичних елементів. У системах, що побудовані з такими законами еволюції, кожному стану $x \in \xi$ відповідає єдина траєкторія – множина

$Y = \{ y \in \xi \mid \exists t_1, t_2, \ldots, t_n,\ y = \Phi_{t_n}(\Phi_{t_{n-1}}(\ldots(\Phi_{t_1}(x)))) \}$. Характеристики траєкторій та їх залежність від параметрів відображень Φ_t є основним об'єктом дослідження теорії динамічних систем.

Динамічні системи можна розділити на два види – дискретні та неперервні. У дискретних системах послідовне застосування відображень Φ_t до початкових умов x дає в результаті дискретну траєкторію, у якій між двома сусідніми точками y_t та y_{t+1} не існує проміжних станів:

$$\begin{aligned} y_{t+1} &= \phi_t(y_t) \\ y_0 &= x. \end{aligned} \tag{1.3}$$

У неперервних системах закон еволюції Φ_t задається диференціальними рівняннями, а траєкторії отримуються шляхом розв'язання задачі Коші для початкових умов x:

$$\begin{aligned} \dot{y}(t) &= \phi_t(y(t)) \\ y(t_0) &= x. \end{aligned} \tag{1.4}$$

Таким чином, у неперервних системах, за умови неперервності та диференційовності закону Φ_t, між будь-якими двома точками траєкторії неперервної системи існує безліч проміжних станів.

З визначення траєкторії випливає, що закон Φ_t до поточного стану системи y_t можна застосовувати нескінченно. Через це поведінка, яку мають траєкторії системи при $t \to \infty$, становить особливий інтерес, причому як для дискретних, так і для неперервних систем. За характером цієї поведінки, динамічні системи можна розділити на два широкі класи – стійкі та нестійкі.

Траєкторії стійких систем з часом збігаються до деяких замкнених підмножин фазового простору ξ, які називають *граничними множинами*. У найпростішому випадку, усі траєкторії стійкої системи збігаються до деякої нерухомої точки y^*, тобто для будь-яких початкових умов $x \in \xi$ існує така траєкторія Y, що $y_{t\to\infty} \to y^*$. Точку y^* називають *ω-граничною* точкою. Якщо гранична множина містить більше однієї точки, а також існують такі $t^* \to \infty$ і $T > 0$, що $y_{t^*} = y_{t^*+T}$, то говорять, що траєкторії замикаються, а у системі присутній *граничний цикл* [32].

Нестійкі системи, на відміну від стійких, не мають глобальних граничних множин, до яких би збігалися усі траєкторії. У найпростішому випадку, траєкторії нестійких систем розбігаються від центру координат і принаймні одна

координата вектора стану у них прямує до нескінченності, тобто $||y_t|| \to \infty$ при $t \to \infty$. У цьому випадку інтерпретація нестійкості та її зв'язок з поняттям граничної множини очевидний. Але існують системи, траєкторії яких залишаються обмеженими ($||y_t|| \leq C$) за будь-яких t, і при цьому не існує таких $t^* \to \infty$, за яких вони збігалися б до граничної множини. З цього випливає, що окрім граничних множин у фазовому просторі системи можуть існувати й інші об'єкти, які здатні притягувати до себе траєкторії. В теорії динамічних систем такі об'єкти (а також самі граничні множини) називають *атракторами*. Якщо закон Φ_t детермінований і виконуються умови $\forall t(\nexists T : y_t = y_{t+T}, \exists C \ll \infty : ||y_t|| \leq C)$, то при $t \to \infty$ траєкторії стають нескінченно щільно упакованими у деякому підпросторі фазового простору ξ, при цьому ніколи його не заповнюючи. Отже, об'єкт, до якого збігаються траєкторії такої системи, може містити нескінченно довгі траєкторії, що не перетинаються, але при цьому не є повноцінним підпростором ξ. Зазвичай це означає, що даний об'єкт є фракталом, а атрактор, який він утворює, називають *дивним*. Формально дивність атрактора можна довести, визначивши для нього *розмірність Гаусдорфа*.

Нехай $\mathbb{R}^d$ – d-вимірний метричний простір з метрикою ρ та для будь-якої підмножини $E \subset \mathbb{R}^d$ функція $diam(E)$ позначає її діаметр.

Означення 1.1. *Нехай $E \in \mathbb{R}^n$. Тоді діаметр E можна позначити як:*

$$diam(E) = \max\{\rho(x, y) : x, y \in E\}.$$

Іншими словами, діаметр множини – це відстань між її найбільш віддаленими одна від одної точками.

Означення 1.2. *Для будь-яких $E \subset \mathbb{R}^d$, $\delta \in [0, \infty]$, $\alpha \in [0, \infty]$ можна задати зовнішню міру розмірності d і обмежену δ:*

$$H^d_\beta(E) = \inf\left\{\sum_{i=1}^{\infty}(diam(E_i)^d) : E \subset \bigcup_{i=1}^{\infty} E_i, diam(E_i) < \delta\right\}. \tag{1.5}$$

Відображення $\delta \to H^d_\delta(E)$ – монотонно незростаюча функція. Тоді d-вимірну міру Гаусдорфа для E можна визначити як:

$$H^d(E) = \lim_{\delta \to 0} H^d_\delta(E) \tag{1.6}$$

Теорема 1.1. *Для* $0 \leq s < t < \infty$ *та* $E \subset \mathbb{R}^d$ *маємо:*

- $H^s(E) < \infty \Rightarrow H^t(E) = 0,$
- $H^t(E) > 0 \Rightarrow H^s(E) = \infty.$

Означення 1.3. *[33] Розмірність Гаусдорфа* $dim_H(E)$ *підмножини* $E \subset \mathbb{R}^d$ *визначається як:*

$$\begin{aligned} \dim_H(E) &= \sup\{s : H^s(E) > 0\} = \sup\{s : H^s(E) = \infty\} \\ &= \inf\{t : H^t(E) = 0\} = \inf\{t : H^t(E) < \infty\}. \end{aligned} \tag{1.7}$$

Враховуючи властивість атракторів притягувати до себе траєкторії, можна сказати, що дослідження конкретної динамічної системи полягає в ідентифікації атракторів, які можуть в ній виникати, а також визначенні їх типів і властивостей.

1.1.1 Класичні методи

Принциповий внесок в розвиток математичного моделювання природних явищ зробив ще у XVIII столітті І. Ньютон, якого через це можна вважати засновником теорії динамічних систем. Саме Ньютон запропонував спосіб для описання природних систем як неперервної послідовності змін їх стану у часі, і формалізував мову для описання цих змін у вигляді диференціальних рівнянь. По-перше, особливості запропонованої ним методики полягали в тому, що у представлення досліджуваного об'єкта включалися не тільки поточні, але й історичні значення компонентів його стану. По-друге, особлива увага приділялася саме взаємозв'язку між двома послідовними станами – наступний стан вважався повністю детермінованим і залежним лише від значень попереднього стану. По-третє, зміни у стані вважались неперервними. Виявилось, що такий підхід до моделювання дозволяє не тільки ясно та просто виражати фундаментальні закони, але і формулювати точні описи навіть найскладніших взаємодій в природних процесах.

В основі сучасного математичного моделювання знаходяться звичайні диференціальні рівняння, які формулюються виходячи з відомих законів фізики, та розв'язуються з метою передбачення стану системи в майбутньому або його реконструювання в минулому.

Розв'язком диференціального рівняння називається деяка функція, яка залежить від часу та початкових даних та задовольняє даному рівнянню. Якщо аналітично розв'язок побудувати не вдається, то використовуються чисельні методи або методи наближеного розв'язку. Ці методи дозволяють досліджувати поведінку системи на скінченному часовому інтервалі за фіксованих початкових умов. Існує багато задач, у тому числі важливих практично, у яких розв'язок диференціальних рівнянь неможливо представити аналітично. Наприклад, для задачі трьох тіл у небесній механіці. Динаміка взаємодії таких тіл сильно залежить від їх мас, взаємного положення у просторі, відносних швидкостей тощо, та може мати безліч стійких та нестійких типів поведінки. Саме тому важливі не просто самі координати взаємодіючих тіл та фактичні значення інших параметрів такої системи, а якісна оцінка системи в цілому та довгострокової еволюції її траєкторій.

Наступним важливим кроком в розвитку моделювання динамічних систем стало створення якісної теорії диференціальних рівнянь. Її основи заклав Анрі Пуанкаре наприкінці XIX – початку XX сторіччя. Головною рисою його методів дослідження став перехід від аналізу кожної з можливих траєкторій та перехід до аналізу простору, в якому вони існують, що дозволяє оцінити поведінку системи лише за її параметрами та початковими умовами та повністю прибирає необхідність у пошуку будь-якого розв'язку. Таким чином головним об'єктом дослідження якісної теорії диференціальних систем став не розв'язок рівняння як функція або числове представлення траєкторії, а фазовий простір ξ, точки якого представляють різні стани системи, заданої законами еволюції Φ_t. Таке трактування задачі аналізу динамічної системи було суто геометричним та дозволило працювати навіть з системами, чиї закони еволюції невідомі – їх траєкторії можна помістити заздалегідь в простір ξ та досліджувати так само як і фазові портрети відомих систем.

Означення 1.4. *Фазовий портрет системи – це геометричне представлення її траєкторій у фазовому просторі, яке графічно зображує їх поведінку.*

Побудова фазових портретів є надзвичайно потужним інструментом дослідження динамічних систем, який дозволяє легко ідентифікувати поведінку їх траєкторій у різних областях фазового простору. Для прикладу можна навести

систему, що описує динаміку математичного маятника:

$$\ddot{\theta} + \frac{g}{l}\sin\theta = 0, \tag{1.8}$$

де g – прискорення вільного падіння, а l – довжина підвісу. За допомогою простої заміни $\dot{\theta} = v$, отримуємо систему з двох звичайних диференціальних рівнянь:

$$\begin{aligned} \dot{\theta} &= v \\ \dot{v} &= -\frac{g}{l}\sin\theta. \end{aligned} \tag{1.9}$$

Для побудови фазового портрета (1.9), знайдемо розв'язки для різних початкових умов в околі початку координат і нанесемо їх на графік, абсцисою якого є значення кута θ, а ординатою – кутова швидкість v. З портрета, наведеного на Рис. 1.1, легко бачити різні типи поведінки траєкторій системи, що можуть виникати залежно від обраних початкових умов. За невеликих початкових значень швидкості v у системі виникають періодичні осциляції, за яких траєкторії обертаються навколо ділянок, що відповідають положенню спокою маятника $\theta = 2\pi n$. Проте якщо початкова швидкість виявиться зависокою, то осциляції не виникнуть і кут буде постійно зростати (спадати), що можна інтерпретувати як постійне обертання маятника в ту чи іншу сторону. Варто відзначити, що фазовий портрет робить очевидним наявність особливих точок у системі – якщо початкове значення швидкості дорівнює нулю, а кут є множником π, система буде перебувати в рівновазі, а її траєкторії не зрушать з місця.

Складні динамічні системи містять багато компонентів, які тісно пов'язані між собою та мають суттєвий вплив одне на одного. Через це кожен компонент системи та його поведінка містять в собі сліди інформації про інші компоненти. Іншими словами, кожна змінна системи несе в собі приховану інформацію про інші її змінні. Це означає, що мають існувати перетворення, які дозволять отримати (наближені) значення невідомих змінних та відновити фазовий портрет досліджуваної системи. Одне з таких перетворень було вперше запропоноване в 1980 році групою американських вчених [34], які показали, що можна отримати задовільну картину дивного атрактора невеликої розмірності, якщо замість змінних x, що входять в рівняння динамічної системи, використовувати m-вимірні вектори, отримані з елементів часового ряду. Роком пізніше була опублікована праця голландського математика Такенса, у якій він сформулював та

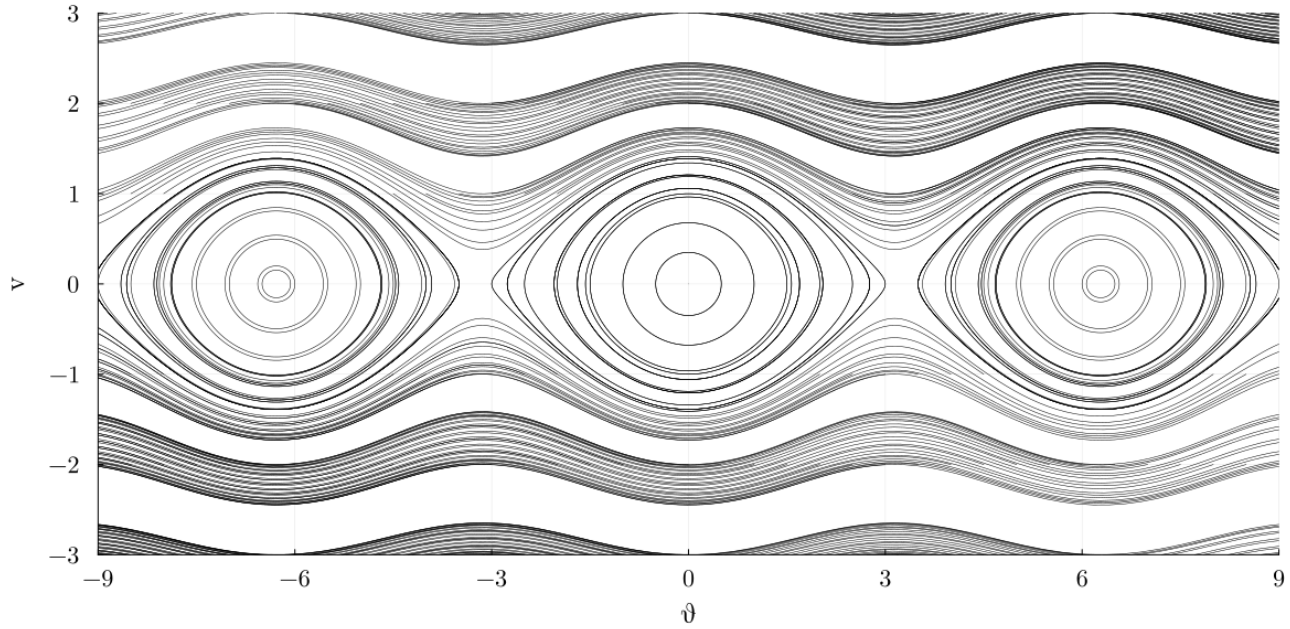


Рис. 1.1: Фазовий портрет динамічної системи (1.9).

довів теорему про зв'язок часового ряду з вимірністю простору вкладення атрактора. Саме ця теорема, огляд якої наведено нижче, лягла в основу більшості алгоритмів аналізу часових рядів як таких, що були породжені динамічними системами.

Нехай задана динамічна система (1.4) з атрактором у фазовому просторі $\xi = \mathbb{R}^m$ та часовий ряд, значення якого є задані деякою скалярною "функцією спостереження" станів системи:

$$U = (u_0, u_1, \ldots, u_n), \tag{1.10}$$

$$u_i = \alpha(y(t_i)). \tag{1.11}$$

Нехай часовий крок між елементами часового ряду дорівнює Δt, а вектори $y(t_i)$ траєкторії Y позначені як y_i. Тоді:

$$\begin{array}{lll} u_i & = \alpha(y_i) & \equiv \Phi_0(y_i) \\ u_{i+1} & = \alpha(y_{i+1}) & \equiv \Phi_1(y_i) \\ \ldots & \ldots & \ldots \\ u_{i+k} & = \alpha(y_{i+k}) & \equiv \Phi_k(y_i). \end{array} \tag{1.12}$$

З (1.12) випливає, що зі значень u_i можна скласти k-вимірний вектор z_i, який можна виразити з одного стану динамічної системи. Введемо векторну

функцію $\Lambda : \mathbb{R}^k \to \mathbb{R}^k$:

$$z_i = (u_t, u_{t-\tau}, \ldots, u_{t-k\tau}) = \Lambda(y_i). \tag{1.13}$$

Теорема 1.2 (Такенса). *[35] Для будь-яких значень $\tau > 0$ і $k \geq 2m + 1$ існує неперервна та ін'єктивна функція Λ.*

Використання теореми Такенса дозволяє будувати реконструкції атракторів динамічних систем з даних часових рядів та досліджувати їх характеристики. На основі цих досліджень можна будувати динамічні моделі, траєкторії яких можна інтерпретувати як прогноз стану досліджуваної системи.

Одним із поширених способів побудови математичної моделі для досліджуваного процесу є *ідентифікація систем.*

Ідентифікація систем (ІС) – це галузь математики, яка займається дослідженням процесу побудови моделей для набору вхідних та вихідних даних. Такі моделі зазвичай будуються у вигляді скінченновимірних систем різницевих або диференціальних рівнянь. ІС тісно пов'язана з іншими галузями, такими як статистика та аналіз часових рядів, та знаходить застосування у розв'язку багатьох прикладних задач [36].

Формально, ІС полягає у побудові відображення множини наборів даних в множину моделей. Прикладом такої моделі є лінійна авторегресійна екзогенна модель (autoregressive exogenous model, ARX), що у загальному вигляді записується як:

$$\begin{aligned} &y(t) + a_1 y(t-1) + \cdots + a_n y(t-n) = \\ &= b_1 u(t-1) + \cdots + b_m u(t-m) + e(t), \end{aligned} \tag{1.14}$$

де y та u – відповідно входи та виходи системи, а e – стохастичний компонент. Більш актуальним для цієї роботи прикладом є модель з неперервним часом, що передбачає наступні стани процесу:

$$\begin{cases} dx(t) = Ax(t)dt + Bu(t)dt + dw(t) \\ dy(t) = Cx(t)dt + Du(t)dt + dv(t), \end{cases} \tag{1.15}$$

де x – вектор внутрішнього стану моделі, а змінні w та v – вінерівські процеси. Компонентами для моделювання нелінійних залежностей в системах (1.15) можуть слугувати будь-які нелінійні функції, в тому числі й штучні нейронні мережі.

Кожна модель (1.15) асоціюється з деякою функцією f, що прогнозує значення $y(t)$ з її дискретизованих минулих значень:

$$\begin{aligned} &\hat{y}(t) = f(Z^{t-1}, t) \\ &Z^{t-1} = \{y(t-1), u(t-1), \ldots, y(0), u(0)\}. \end{aligned} \tag{1.16}$$

Множина гладко параметризованих функцій $f(Z^{t-1}, t, \Theta)$ утворюють структуру моделі $\mathcal{M}$ з параметрами Θ з множини $D_{\mathcal{M}}$ розмірністю d. Відображення з наявних даних Z^N в простір можливих параметрів $D_{\mathcal{M}}$, що дозволяє обрати Θ, можна побудувати за допомогою методів оптимізації, що мінімізують відхилення прогнозу моделі від реальних значень:

$$\hat{\Theta}_N = \text{argmin}_{\Theta \in D\mathcal{M}} \sum_{\mathcal{M}}^{N_t=1} \mathcal{L}\left(y(t) - f(Z^{t-1}, t, \Theta)\right), \tag{1.17}$$

де $\mathcal{L}$ – скалярна невід'ємна функція.

Підвидом ІС є адаптивна ідентифікація систем (AIC), або рекурсивна (покрокова) ідентифікація. Її можна виразити як додаткові обмеження, що накладаються на модель (1.16) та її параметри (1.17):

$$\begin{aligned} &\hat{\Theta}_N = h(X_N) \\ &X_N = H(N, X_{N-1}, y(N), u(N)), \end{aligned} \tag{1.18}$$

де $X(t)$ – вектор з фіксованою вимірністю. Така структура дозволяє отримувати прогноз стану в момент N за сталу кількість обчислень, що дозволяє подібним моделям оброблювати потоки даних в реальному часі, коли повний набір даних модельованого часового ряду недоступний. Завдяки цьому на основі моделі (1.18) можна будувати адаптивний контроль, адаптивну фільтрацію, нагляд, та інше. Її структура часто набуває вигляду:

$$\begin{cases} \hat{\Theta}_N &= \hat{\Theta}_{N-1} + \gamma(N) Q_1(X(N), y(N), u(N)) \\ X_N &= X_{N-1} + \mu_N Q_2(X_{N-1}, y(N), u(N)), \end{cases} \tag{1.19}$$

що дозволяє більш ясно відобразити адаптування поточних параметрів залежно від стану системи. У цьому випадку γ та μ виступають мета параметрами та зазвичай залежать від часового кроку між значеннями в часовому ряді.

1.1.2 Нейромережевий підхід

Теорія динамічних систем надає інструменти для формалізації досліджень динамічних процесів та створення їх математичних моделей. Традиційно такі моделі конструюють "вручну" шляхом виведення диференціальних рівнянь, що задають динаміку процесу, із фундаментальних законів фізики. Побудовані таким чином моделі мають велику пояснювальну силу, але вони часто нехтують змінними, які не грають головну роль у визначенні динаміки процесу. Наприклад, модель (1.9) повністю ігнорує розміри тіла, деформацію підвісу та тертя в точці підвісу. Однак прикладні задачі часто вимагають точного опису складних систем, який при цьому ще є легким та швидким в обчисленні. Саме тому набрала великої популярності альтернативна парадигма моделювання, метою якої є автоматична апроксимація динаміки процесу безпосередньо із наявних даних шляхом оптимізації деякої скалярної функції втрат $\mathcal{L}$, що дає кількісну оцінку точності обраної моделі для значень вхідних та цільових даних відносно поточної множини параметрів Θ. Одним з найбільш поширених інструментів для такої апроксимації є різні форми штучних нейронних мереж, які використовуються, зокрема, для ідентифікації систем, машинного навчання, та глибокого навчання [37].

Штучна нейронна мережа – це один зі способів реалізації штучного інтелекту (ШІ), в основі якого лежать спроби імітації роботу мозку людини. Нейрон є особливою біологічною клітиною, яка обробляє та передає інформацію. Він складається з тіла (соми), і відростків нервових волокон двох типів: дендритів, які приймають імпульси, і єдиного аксона, по якому нейрон може передавати імпульс. Тіло нейрона містить ядро, яке містить інформацію про спадкові властивості, і плазму, що володіє молекулярними засобами для утворення необхідних для нейрона речовин. Таким чином, нейрон отримує сигнали (імпульси) від аксонів інших нейронів через дендрити (приймачі) і передає сигнали, згенеровані тілом клітини, уздовж свого аксона (передавача), який в кінці розгалужується на волокна. На закінченнях цих волокон знаходяться спеціальні утворення – синапси, які впливають на величину імпульсів. Кожен нейрон може знаходитися в стані збудження або гальмування.

Штучний нейрон в першому наближенні імітує властивості біологічного нейрона. Сигнали, що отримуються дендритами від аксонів інших нейронів, подаються у вигляді вектора числових значень $\{x_1, x_2, \ldots, x_n\}$. Внутрішні процеси

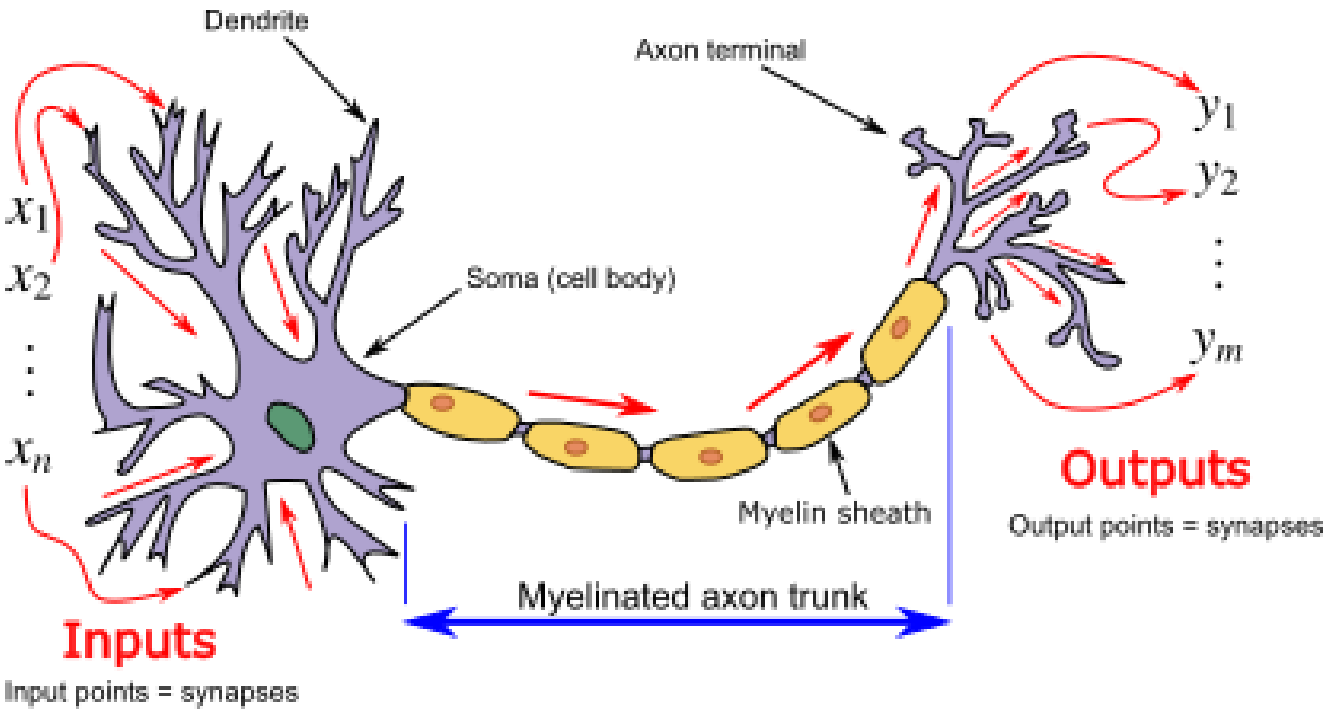


Рис. 1.2: Структура біологічного і штучного нейронів та зв'язок між ними [38].

в нейроні, що визначають потужність вихідного сигналу, моделюються як множення вхідного вектора на вектор вагових коефіцієнтів $\{w_1, w_2, \ldots, w_n\}$ та застосуванням до результату деякої нелінійної функції активації σ. Таким чином, кожен коефіцієнт w_i задає силу синаптичного зв'язку даного нейрона з одним з попередніх нейронів, а функція активації забезпечує можливість моделювання нелінійних зв'язків між вхідними сигналами. Приклади деяких функцій активації наведено в Табл. 1.1. Графічне зображення будови штучного нейрона наведено на Рис. 1.2.

Назва	Функція	Область значень
Порогова (одиничного стрибка)	$\sigma(x) = \begin{cases} 0, s < 0 \\ 1, s \geq 0 \end{cases}$	$[0, 1]$
Степенева	$\sigma(x) = \begin{cases} -\delta\lvert x\rvert^{\alpha}, s < 0 \\ \beta\lvert x\rvert^{\alpha}, s \geq 0 \end{cases}$	$(-\infty, \infty)$
Логістична (сигмоїда)	$\sigma(x) = \dfrac{1}{1 + e^{-\alpha x}},\ \alpha > 0$	$(0, 1)$
Гіперболічний тангенс	$\sigma(x) = \dfrac{e^{\alpha x} - e^{-\alpha x}}{e^{\alpha x} + e^{-\alpha x}}\ \alpha > 0$	$(0, 1)$
Зрізаний лінійний вузол	$\sigma(x) = \dfrac{x + \lvert x\rvert}{2}$	$[0, \infty)$

Табл. 1.1: Приклади функцій активації, що можуть застосовуватися в нейромережевих моделях.

Потужність нейромережевого підходу в моделюванні досягається об'єднанням нейронів у мережі, які зазвичай мають багатошарову структуру, як показано в співвідношеннях (0.2), та використовують нелінійну функцію активації σ. У такому випадку кожен шар моделі містить багато нейронів, а оскільки кожен нейрон моделюється як зважена сума (див. рівняння (0.1)), шар представляється як множення вектора входів на матрицю вагових коефіцієнтів W. Таким чином, елемент вагової матриці w_{ij} описує ступінь впливу вихідного сигналу j-го нейрона попереднього шару на i-й нейрон поточного шару.

Найпростіше застосування багатошарової нейронної мережі вигляду (0.2) є побудова моделі для відображення точок деякого простору X в точки іншого простору Y. Наприклад, X може бути множиною зображень рукописних цифр, а Y – множиною їх ідентифікаторів (лейблів) від 0 до 9. Розв'язання саме цією задачі стало одним з перших важливих досягнень галузі нейромережевого моделювання [39], а база даних зображень рукописних цифр та їх лейблів MNIST [40] досі широко використовується для валідації нових розробок в галузях комп'ютерного зору та порівнянні різних методів глибокого навчання для розпізнання образів [40].

У контексті динамічних систем здатність навіть простих нейромереж апроксимувати будь-які гладкі нелінійні відображення $\mathbb{R}^n \to \mathbb{R}^m$ [41] може бути використана для моделювання значень похідних у фазовому просторі моделі (1.3) або (1.4).

Оскільки нейронні мережі є моделями, що отримуються безпосередньо з даних, а не виводяться із законів фізики для конкретного процесу, вони не мають специфічною для даного процесу поведінки до завершення тренування. Через це важливо не тільки як нейромережа обчислює результат, але і яким чином змінюються її параметри під час тренування. Для простих нейромереж тренування полягає у розв'язанні задачі оптимізації відносно параметрів моделі $\Theta = \{W_i, b_i\},\ \forall i$ для такої функції втрат $\mathcal{L}$, яка задає середньоквадратичне відхилення (mean square error, MSE) результату моделі $\hat{y}$ від істинних значень y, які вона має апроксимувати:

$$\mathcal{L}(x, y, \Theta) = \frac{1}{m}||y - F(x, \Theta))||_2. \tag{1.20}$$

Проте для моделі, яка апроксимує динамічну систему, функція втрат має дещо

складнішу структуру:

$$\begin{aligned} &\mathcal{L}(x, y, \Theta) = \frac{1}{nm} \sum_{i=1}^{n} \sum_{j=1}^{m} (y_{ij} - \hat{y}_{ij})^2 \\ &\hat{y} = F(x, \Theta)), \end{aligned} \tag{1.21}$$

де значення y – цільова траєкторія, $\hat{y}$ – траєкторія, породжена динамічною системою, заданою нейромережею $F(x, \Theta)$, n – довжина траєкторії, а m – розмірність фазового простору системи.

Існують два основні підходи до моделювання динамічних систем за допомогою нейронних мереж – *прямі* та *покрокові*. Прямі моделі не використовують визначені інтеграли для обчислення результату і натомість намагаються передбачити стан системи в деякий час t_i шляхом включення цієї часової позначки у вектор вхідних даних. Покрокові моделі використовують чисельні методи для обчислення траєкторії системи та прогнозування її майбутніх станів шляхом покрокового інтегрування визначених нею рівнянь. Головна різниця між цими двома підходами у їх здатності працювати з різними початковими даними та входами. За дизайном, покрокові моделі здатні працювати з різними початковими умовами, у той час, як прямі моделі потребують повторного тренування. Іншими словами, покрокові моделі здатні вивчити динаміку, а прямі моделі вивчають лише розв'язок задачі Коші для початкового стану та набору вхідних параметрів. Розглянемо обидва класи моделей детальніше.

Перший спосіб побудови апроксимації траєкторій динамічної системи є моделювання прямої залежності її стану $y \in \xi$ від часу t, через що такі моделі й називаються прямими. Щоб побудувати таку модель, зазвичай тренують нейромережу, щоб та могла надавати точні значення розв'язку на деякому наборі *точок колокації*, значення яких відібрані із модельованої системи. Іншими словами, нейронна мережа у цьому випадку виступає як інтерполяційний рушій, який дозволяє знаходити значення системи поза інтерполяційними точками. Важливо зазначити, що суттєвим обмеженням таких моделей є те, що вони тренуються лише для конкретного набору початкових умов. Щоб мати змогу застосовувати модель для системи з іншими початковими умовами, модель має бути натренована заново.

Такий тип моделей зазвичай застосовується для інтерполяції розв'язків диференціальних рівнянь в частинних похідних, та рідше – звичайних систем диференціальних рівнянь. Це зумовлено декількома причинами. По-перше, рівня-

ння в частинних похідних зазвичай мають високу складність та знаходження їх розв'язку часто вимагає суттєвих обчислювальних ресурсів, у той час, як нейронна мережа вигляду (0.2) містить прості операції, такі як множення матриць, обчислення яких на сучасних комп'ютерах є надзвичайно швидким завдяки ефективним алгоритмам [42]. По-друге, багато практичних застосувань звичайних диференціальних рівнянь передбачає обчислення розв'язків для багатьох різних початкових умов, що означає необхідність у частому повторному тренуванні нейромережі.

Другий спосіб побудови динамічної системи на основі нейромережі – використання покрокових моделей. Для динамічних систем, заданих системами звичайних диференціальних рівнянь, головною задачею є виведення аналітичного вигляду цих рівнянь, на яких потім можна застосувати чисельні методи інтегрування для отримання траєкторій системи. Прикладом такого підходу є модель (1.9) та її траєкторії на Рис. 1.1. При застосуванні нейромережевого моделювання, нейронна мережа апроксимує істинні похідні модельованої системи так, щоб її можна було використати замість аналітичного представлення під час чисельного інтегрування. Такі моделі називаються покроковими, що відображає спосіб роботи чисельних інтеграторів, які обчислюють стани системи на багато кроків вперед у часі. Перевага таких моделей у тому, що вони можуть використовувати різні види інтеграторів, які найкраще підходять для роботи з конкретним типом динаміки, або взагалі не використовувати схеми інтегрування.

Вибір інтегратора у покрокових моделях є настільки ж важливим, як і структура самої моделі, оскільки він є незалежним від нейромережі та може суттєво впливати на її точність, швидкість обчислення, тип результату, та навіть критерій оптимізації.

У найпростішому випадку наступний стан системи передбачається моделлю напряму, без використання інтеграторів:

$$y_{k+1} = F(y_k, \Theta), \tag{1.22}$$

що еквівалентно означенню дискретних динамічних систем (1.3).

Нейронні мережі (1.22) тренуються так, щоб давати задовільний прогноз наступного стану y_{k+1} залежно від поточного стану y_k. Під час тренування, ця операція векторизується, щоб усі стани, окрім останнього, обчислювалися мо-

деллю одночасно. Таким чином функція втрат $\mathcal{L}$ зможе оцінити усі відхилення від бажаних значень одразу та надати більш коректний градієнт для оптимізації, а її виклики будуть більш економними. Під час застосування моделі, їй на вхід подаються лише початкові умови x, а прогноз траєкторії $y_1, y_2, \ldots, y_k$ отримується шляхом рекурсивного застосування моделі до власних результатів, тобто

$$\begin{aligned} y_1 &= F(x, \Theta) \\ y_2 &= F(y_1, \Theta) \\ &\ldots \\ y_k &= F(Fy_{k-1}, \Theta). \end{aligned} \tag{1.23}$$

Другий клас покрокових моделей використовує залишковий підхід, що тренує модель передбачати не сам наступний стан, а наближення його похідної в поточній точці:

$$y_{k+1} = y_k + F(y_k, \Theta). \tag{1.24}$$

Модель (1.24) в галузі машинного навчання називається залишковою нейромережею (residual network, ResNet), що здатна ефективно розв'язувати різноманітні задачі, починаючи з класифікації зображень і закінчуючи обробкою мовлення. Також це рівняння є майже ідентичним формулі чисельного інтегрування методом Ейлера з часовим кроком h:

$$y_{k+1} = y_k + hf(t_k, y_k). \tag{1.25}$$

Єдина відмінність у тому, що залишкова мережа не дозволяє вказувати часовий крок h і натомість масштабує прогноз похідної залежно від тренувальних даних.

Нарешті, останній клас розвиває головну ідею моделі (1.24) та використовує її схожість на схему інтегрування Ейлера. Нейронні звичайні диференціальні рівняння (neural ODEs, NODEs) [43] моделюють дані як неперервний потік та використовують більш точні методи чисельного інтегрування для його апроксимації:

$$y_{k+1} = y_k + \int_0^T F(y_k, \Theta, t)\, dt, \tag{1.26}$$

Такий підхід до моделювання дозволяє застосовувати класичні методи для оцінки та гарантування стабільності траєкторій, такі як теорія стійкості Ляпу-

нова. Проте недоліком моделей на базі NODE є висока обчислювальна складність у порівнянні з іншими моделями, оскільки для отримання результату моделі необхідно чисельно інтегрувати потенційно складну систему ЗДР.

1.2 Обробка сигналів

У попередньому розділі обговорювались різноманітні математичні системи та методи їх побудови для моделювання поведінки природних процесів. При цьому наявність числових даних у вигляді (1.10) та зручний доступ до них вважалися даністю. Однак без цього критичного компонента увесь процес моделювання стає неможливим, а некоректне перетворення стану процесу на послідовність значень може суттєво вплинути на якість моделі. Тому перш ніж переходити безпосередньо до обговорення практичних задач моделювання, розглянемо поняття сигналів та способи роботи з ними.

Навколишнє середовище наповнене різноманітними сигналами, причому як природними, так і створені людьми. Прикладом сигналу є зміни повітряного тиску, що ми сприймаємо як звук, щоденні коливання температури, періодичні електричні розряди у м'язах серця. Сигнали є джерелами інформації, проте часто ця інформація виявляється прихованою або зашумленою. Підвищення інформативності сигналів, а також вилученням, зберіганням, та передачею прихованої в них інформації займається така галузь прикладної математики як *обробка сигналів*. Найчастіше розглядаються електричні форми представлення сигналів, оскільки вони забезпечують найбільшу гнучкість при виконанні завдань цієї галузі.

Значення будь-якого сигналу у кожен момент часу визначаються його (миттєвою) амплітудою. При цьому час може бути як неперервним t, так і дискретним $n\Delta t$, де Δt – це інтервал *дискретизації*, а n – ціле число. Амплітуда також може або набувати неперервних значень, або бути *квантованою* на скінченну кількість інтервалів між її найменшим та найбільшим значеннями. Різні комбінації застосування дискретизації та квантування наведено на Рис. 1.3.

Музика, що відтворюється колонками комп'ютера, є прикладом *аналогового сигналу*, оскільки звук, що ми чуємо, є результатом неперервної зміни повітряного тиску. Проте інформація про цей звук зберігається на комп'ютері у цифровому вигляді – щоб бути почутою, вона має бути перетворена на аналоговий

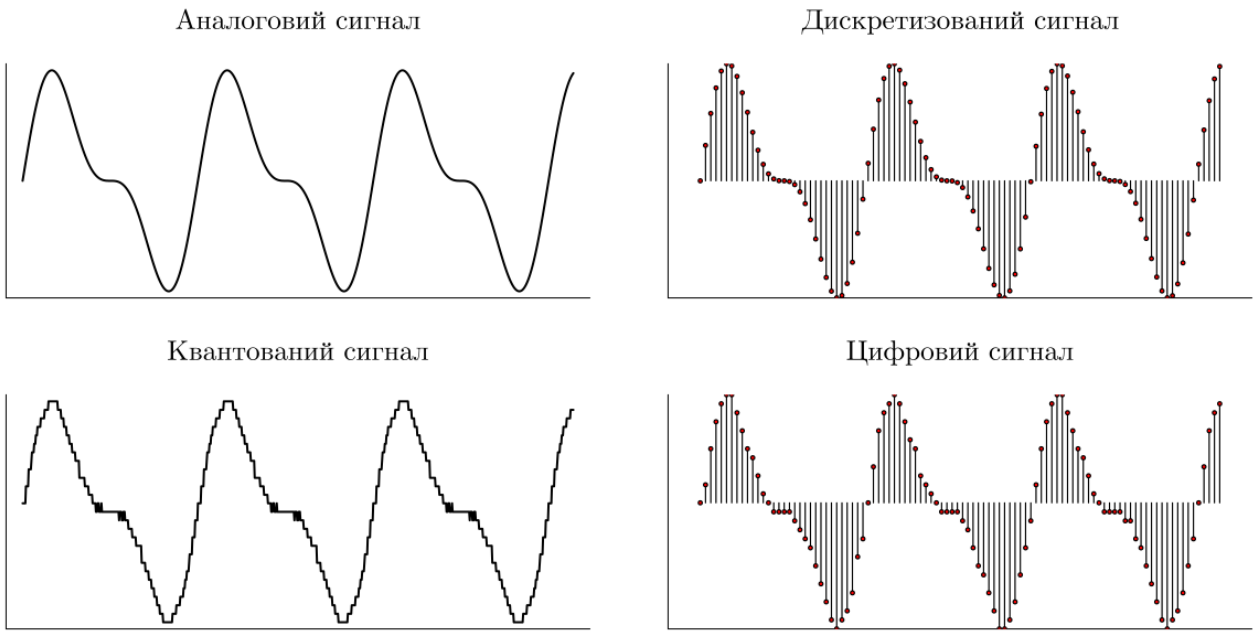


Рис. 1.3: Аналоговий, дискретизований, квантований, та цифровий сигнали.

сигнал. Щорічна інформація про кількість населення є прикладом *дискретного сигналу*, оскільки вона дискретизована у часі (інтервал дискретизації Δt тут рівний одному року) та вимірюється у квантованих одиницях (людях).

Аналогові сигнали були головним предметом дослідження у минулому, проте з широким розповсюдженням комп'ютерів, фокус досліджень сильно змістився до роботи з цифровими сигналами. Оскільки вони є просто послідовністю чисел, для їх обробки можна використовувати такі ж логічні схеми, що використовуються у цифрових комп'ютерах.

Цифрова обробка аналогових сигналів потребує використання аналогово-цифрових перетворювачів (АЦП), що дискретизує та квантує вхідний сигнал, та цифро-аналогових перетворювачів (ЦАП), що перетворює оброблений сигнал назад до аналогової форми.

Процес дискретизації часто є необхідним з практичної точки зору та є першим кроком у *цифровій обробці сигналів* (ЦОС). Другим кроком є квантування. Кожен з цих кроків може призвести до втрати інформації. Втрат можна здебільшого уникнути, якщо скористатися теоремою відліків та визначити максимальну частоту F_{max}, що становить інтерес у даній задачі, та дискретизувати дані з кроком не більше за $\dfrac{1}{2 \cdot F_{max}}$. Проте більшість сигналів не обмежені ніякими частотами й навіть найменші періоди дискретизації Δt можуть виявитися для них завеликими. Якщо частота дискретизації перевищує критичне значення

$\frac{1}{2 \cdot F_{max}}$, в оцифрованому сигналі може виникнути аліасинг – спектральне спотворення, що полягає у маскуванні високих частот та їх накладанні на низькі частоти. Аліасинга в сигналі неможливо позбутися після завершення процесу дискретизації. Саме тому аналоговий сигнал необхідно обробити фільтром низьких частот, що обмежує його спектральний вміст та видаляє небажані високочастотні компоненти. Приклад аліасингу наведено на Рис. 1.4.

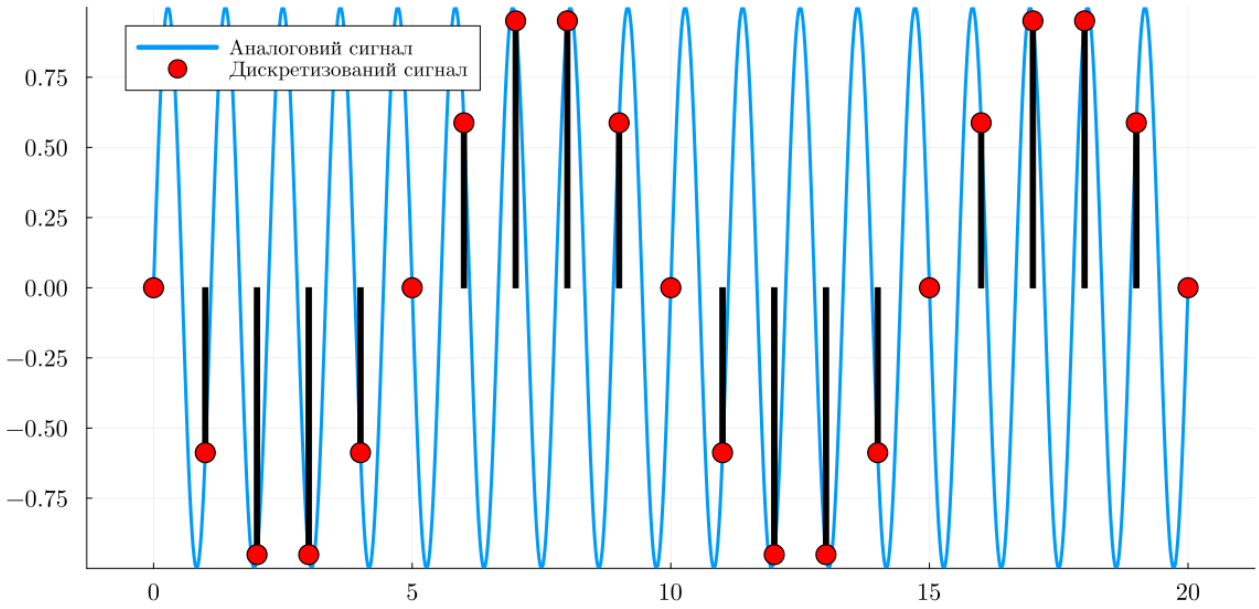


Рис. 1.4: Спотворення оригінального (аналогового) сигналу з частотою, що значно перевищує частоту дискретизації.

Теорема 1.3. *(Віттекера – Найквіста – Котельникова – Шеннона). [44] Якщо неперервний сигнал $u(t)$ має спектр, обмежений частотою F_{max}, то його можна однозначно і без втрат відновити за дискретними відліками, узятими з частотою $f_s = 2F_{max}$, або, по-іншому, за відліками, взятими з періодом* $\Delta t = \frac{1}{2 \cdot F_{max}}$

Означення 1.5. *Частота Найквіста – найвища частота, яка система здатна обробити без спотворень.*

Оскільки обробка чисельних даних на цифрових комп'ютерах вимагає скінченних даних зі скінченною точністю, амплітудам дискретизованого сигналу необхідно призначити деяке значення зі скінченної множини рівнів. Цей процес називається квантуванням. Його вплив на дані може бути нелінійним та піддаватися лише статистичному опису. Квантування веде до незворотної втрати

інформації та зазвичай виконується як останній крок в системах, що покладаються на АЦП [44].

1.2.1 Аналогові сигнали

Вивчення сигналів дозволяє визначити найбільш ефективні способи вилучення з них інформації. Саме це і є головною задачею цієї галузі математики. Аналоговий сигнал може бути описаний математичним виразом, зображений графічно у вигляді кривої, або навіть представлений у вигляді таблиці значень. Проте реальні сигнали рідко можливо описати кількісно. Часто їх можна лише апроксимувати моделями, до яких можна застосовувати математичні перетворення.

Сигнали можуть бути скінченними або нескінченними. Скінченні сигнали називають *обмеженими за часом*. Напівнескінченні сигнали називають *правосторонніми* (*лівосторонніми*), якщо їх амплітуда рівна нулю для усіх t менших (більших) за деяке скінченне значення α. Сигнали, які рівні нулю для $t < 0$ називають *причинними*. *Частково неперервні сигнали* визначаються різними виразами на різних часових інтервалах. Неперервні сигнали, такі як $u = sin(t)$, визначаються одним виразом для усіх значень t. *Періодичні сигнали* – це нескінченні сигнали, для яких існує таке значення T (період), що $\forall n \forall t : u(t) = u(t+nT)$. Односторонні або обмежені за часом сигнали не можуть бути періодичними.

Корисною метрикою розміру сигналу є його площа. Сигнал $u(t)$ називається абсолютно інтегровним, якщо він має скінченну абсолютну площу:

$$\int_{-\infty}^{\infty} |u(t)|dt < \infty. \tag{1.27}$$

Усі обмежені за часом сигнали зі скінченими амплітудами мають скінченну абсолютну площу. Критерій абсолютної інтегровності часто використовується для перевірки стабільності системи та обґрунтування існування певних перетворень.

Інтерпретація інтеграла квадрата сигналу $u^2(t)$ прив'язана до кількості енергії, отримуваної резистором з опором 1 Ω. *Миттєва потужність* (у ватах) для такого резистора записується як $p_i(t) = u^2(t)$, де u – напруга на ньому або струм через нього. Повна енергія E, отримана 1-Ω резистором, називається *енергією*

сигналу та може бути обчислена шляхом інтегрування миттєвої потужності:

$$E = \int_{-\infty}^{\infty} p_i(t)dt = \int_{-\infty}^{\infty} |u(t)|^2 dt. \tag{1.28}$$

Використання абсолютних значень $u(t)$ в (1.27) та (1.28) дозволяє цим визначенням бути застосовними й для сигналів, що задаються комплексними функціями.

Потужність сигналу P дорівнює середньому значенню енергії сигналу за весь час. Якщо $u(t)$ – періодичний сигнал з періодом T, його потужність визначається середньою енергією за один період:

$$P = \frac{1}{T} \int_{T} |u(t)|^2 dt. \tag{1.29}$$

Ще однією важливою характеристикою сигналів є квадратний корінь його потужності та позначається rms:

$$u_{\text{rms}} = \sqrt{P}. \tag{1.30}$$

Значення $\sqrt{P}$ відповідає постійному струму з такою ж потужністю, що і сигнал $u(t)$. Для сигналів середнє значення u_{av} ніколи не може перевищувати значення u_{rms}, тобто $u_{\text{av}} \leq u_{\text{rms}}$. Ця нерівність є корисною для оцінки потужностей простих періодичних сигналів, що описуються синусоїдами або комплексними експоненційними функціями:

$$u(t) = A\cos(2\pi F_0 t + \theta) \;\Rightarrow\; P = 0.5A^2,$$

$$u(t) = Ae^{\pm i(2\pi F_0 t + \theta)} \;\Rightarrow\; P = A^2.$$

Якщо $u(t)$ – неперіодичний сигнал, його потужність та середнє значення обчислюються на обмеженому відрізку часу довжиною T_0, який можна спрямувати до нескінченності та отримати:

$$P = \lim_{T_0 \to \infty} \frac{1}{T_0} \int_{T_0} |u(t)|^2 dt, \tag{1.31}$$

$$u_{\text{av}} = \lim_{T_0 \to \infty} \frac{1}{T_0} \int_{T_0} u(t)dt. \tag{1.32}$$

Визначення енергії та потужності дозволяє класифікувати сигнали за обома цими показниками. Сигнали зі скінченною енергією називаються *енергетичними*, або *квадратно інтегровними*. Енергетичні сигнали мають нульову потужність, оскільки їх скінченна енергія розподілена по усьому (нескінченному) часу. Усі обмежені за часом сигнали зі скінченними значеннями амплітуди є енергетичними. Прикладами енергетичних сигналів також є односторонні або двосторонні затухаючі або приглушені експоненти або синусоїди. Формально, $u(t)$ вважається енергетичним, якщо всі його амплітуди скінченні та значення $u^2(t)$ прямують до нуля швидше за $\frac{1}{|t|}$ при $|t| \to \infty$.

Сигнали називають *потужніснимu*, якщо вони мають скінченну потужність, що не рівна нулю. Енергія потужнісних сигналів нескінченна. Усі періодичні сигнали та їх комбінації є потужнісними.

Окрім комбінування сигналів між собою, над ними можна проводити ще ряд операцій, які змінюють їх амплітудні або часові характеристики.

Масштабування амплітуди $u(t)$ полягає у множенні усіх його значень на деяку константу C для отримання нового сигналу $Cu(t)$.

Зсув амплітуди додає константу K до кожного (навіть нульового) значення $u(t)$. Результатом зсуву є новий сигнал $K + u(t)$.

Масштабування часу прискорює або сповільнює час, що стискає або розтягує сигнал. Сигнал $v(t) = u(t/2)$ описує подвійне розширення сигналу $u(t)$, оскільки час у ньому сповільнюється удвічі. Так само сигнал $v(t) = u(3t)$ стискає втричі оригінальний сигнал, прискорюючи час в три рази. У загальному вигляді такі перетворення можна записати як $v(t) = u(\alpha t)$

Часовий зсув зміщує сигнал $u(t)$ у часі не змінюючи його форми. Нехай сигнал $v(t) = u(t - \beta)$. Значення $v(t)$ в точці $t = \beta$ дорівнює значенню $u(t)$ в $t = 0$. Іншими словами, сигнал $v(t)$ є сигналом $u(t)$ (зсунутим вправо на графіку) з деякою затримкою β. Так само сигнал $v(t) = u(t + \beta)$ є (зсунутим вліво на графіку) висунутим уперед на час β оригінальним сигналом.

Відображення або *складання* можна виразити у термінах операції масштабування часу з параметром $\alpha = -1$. В результаті цієї операції отримується сигнал $v(t) = u(-t)$

Усі ці операції можна комбінувати для отримання нових сигналів у вигляді $v(t) = K + Cu(\alpha t + \beta)$, а особливо корисні ці операції виявляються при роботі із синусоїдами.

Синусоїди та синтезовані з них гармонічні сигнали є найбільш корисними періодичними сигналами. У загальному випадку синусоїда описується формулами:

$$\begin{aligned} u_p(t) &= A\cos(2\pi F_0 t + \theta) \\ u(t) &= Ae^{i(2\pi F_0 t+\theta)}, \end{aligned} \tag{1.33}$$

що пов'язані тотожністю Ейлера:

$$u_p(t) = \mathrm{Re}\left\{Ae^{i(2\pi F_0+\theta)}\right\} = 0.5Ae^{i(2\pi F_0+\theta)} + 0.5Ae^{-i(2\pi F_0+\theta)}. \tag{1.34}$$

Якщо позначити кутову частоту як $\omega_0 = 2\pi F_0$ і записати $u_p(t) = A\cos(\omega_0 t + \theta) = A\cos[\omega_0(t - t_p)]$, то величина $t_p = -\dfrac{\theta}{\omega_0}$ – це *фазова затримка*, що описує затримку у часі в сигналі, що залежить від фазового зсуву θ. Зазначимо, що аналогові синусоїди та гармонічні сигнали завжди періодичні та унікальні відносно обраного періоду або частоти.

Спільним періодом або просто *періодом* комбінації синусоїд називається найменший проміжок часу за який кожна синусоїда завершує ціле число циклів. Він задається найменшим спільним кратним наявних у сигналі періодів. *Фундаментальна частота* F_0 – це обернене до спільного періоду значення та дорівнює найбільшому спільному множнику наявних у сигналі частот. Спільний період (частоту) можна знайти для будь-яких періодичних сигналів, чиї періоди (частоти) задані раціональними числами. Якщо одна із синусоїд у сигналі має нераціональне значення частоти, спільну частоту знайти неможливо, оскільки такий сигнал ніколи не повторює сам себе. Такі сигнали називаються *майже періодичними* або *квазіперіодичними*.

Для сигналу, заданого комбінацією синусоїд з різними частотами, потужність P обчислюється як сума потужностей окремих синусоїд, а середньоквадратичне значення обчислюється як $\sqrt{P}$.

Важливість гармонічних сигналів та синусоїд полягає у тому, що їх комбінаціями можна представити будь-який сигнал: періодичний – дискретними частотами у вигляді ряду Фур'є, квазіперіодичний – комбінацією усіх можливих значень частот.

1.2.2 Дискретні сигнали

Дискретні сигнали можуть виникати природно, якщо вимірювана величина не є неперервною (наприклад, значення чисельності популяції), або бути результатом дискретизації аналогових сигналів (зазвичай, з постійним кроком Δt). Дискретний сигнал $u(n)$ – це впорядкована множина (ряд) значень, індекси яких описують історію сигналу. Значення також називають *семплами*, а процес їх отримання з аналогового сигналу – *семплуванням*. У визначенні дискретного сигналу не міститься інформації про фактичний інтервал дискретизації Δt, з яким його було отримано, проте її можна вивести, якщо зіставити індекси семплів з відповідними їм моментами у часі. На графіку дискретний сигнал виглядає як послідовність вертикальних ліній на значеннях індексів.

Міри енергії (1.28) та потужності (1.29), визначені для аналогових сигналів, також існують і для дискретних. Для їх виведення необхідно лише замінити операцію інтегрування оператором обчислення суми:

$$E = \sum_{n=-\infty}^{\infty} |u(n)|^2, \tag{1.35}$$

$$P = \frac{1}{N} \sum_{n=0}^{N-1} |u(n)|^2. \tag{1.36}$$

Визначення операцій над дискретними сигналами здебільшого еквівалентні їх визначенням для аналогових сигналів:

- Масштабування амплітуди – $v(n) = Cu(n)$
- Зсув амплітуди – $v(n) = K + u(n)$
- Масштабування часу – не визначено, оскільки інформація про значення амплітуди між двома сусідніми значеннями відсутня.
- Часовий зсув – $v(n) = u(n + \beta)$
- Відображення/складання – $v(n) = u(-n)$

Альтернативами до операції масштабування часу для дискретних сигналів є *децимація* та *інтерполяція*, результатом яких є зміна кількості значень у сигналі.

Нехай $u(n)$ містить значення аналогового сигналу $u(t)$, що були отримані з інтервалом дискретизації Δt. Тоді сигнал $v(n) = u(2n)$ має містити значення стиснутого сигналу $u(2t)$, дискретизовані з тим же інтервалом Δt, або, у термінах $u(n)$, – значення, що відповідають парним індексам $u(n)$ ($v(0) = u(0)$, $v(1) = u(2)$, ...). Ці ж значення можна отримати шляхом дискретизації $u(t)$ з інтервалом $2\Delta t$, що своєю чергою означає зменшення частоти дискретизації вдвічі. Децимація порядку N еквівалентна семплуванню сигналу $u(t)$ з інтервалами $N\Delta t$, що зменшує частоту дискретизації в N разів. Децимований сигнал $u(Nn)$ може бути отриманий з $u(n)$ шляхом вибору кожного N-го значення (тобто значень з індексами $k = Nn$) та відкиданням решти значень.

Якщо $u(n)$ – дискретизований сигнал $u(t)$ з інтервалом Δt, то $v(n) = u(n/2)$ – дискретизований сигнал $u(t)$ з інтервалом $\Delta t/2$, що містить вдвічі більше значень ніж $u(n)$, по одному між кожною парою вже наявних семплів. Якщо сигнал $u(t)$ відомий, то нові значення можна було б отримати безпосередньо з нього. Проте якщо відомі лише значення з $u(n)$, єдиним способом для побудови $v(n)$ є інтерполяція. Наприклад, можна вважати усі нові невідомі значення нульовими (нуль-інтерполяція), рівними попередньому значенню (крокова інтерполяція), або знаходити середнє значення двох сусідніх семплів (лінійна інтерполяція). Нуль-інтерполяція називається *підвищенням дискретизації* та грає важливу роль у практичних методах інтерполяції. Інтерполяція порядку N еквівалентна дискретизації сигналу $u(t)$ з інтервалом $\Delta t/N$, що збільшує частоту дискретизації та кількість семплів у сигналі в N разів.

Зауважимо, що інтерполяція та децимація не є оберненими операціями. Єдиний випадок, коли значення оригінального сигналу $u(n)$ можна відновити, це послідовне застосування інтерполяції, а потім децимації з однаковим порядком N. Розглянемо як приклад дискретний сигнал:

$$\{1, 2, 6, 4, 8\},$$

та застосуємо до нього децимацію та інтерполяцію:

$$\{1, 2, 6, 4, 8\} \underset{n\to 2n}{\to} \{1, 6, 8\} \underset{n\to n/2}{\to} \{1, 1, 6, 6, 8, 8\}. \tag{1.37}$$

Як видно, значення 2 та 4 було втрачено. У загальному випадку, підібрати схему інтерполяції для точного відновлення втрачених під час децимації даних немо-

жливо, тому якщо обидві ці операції необхідно застосовувати одна за одною, рекомендується спочатку використовувати інтерполяцію:

$$\{1,2,6,4,8\} \underset{n\to n/2}{\to} \{1,6,8\} \underset{n\to 2n}{\to} \{1,2,6,4,8\}. \tag{1.38}$$

Отже, децимація є оберненою до інтерполяції, але не навпаки.

Властивості інтерполяції та децимації дозволяють визначити масштабування часу для дискретних сигналів в термінах збільшення та зменшення кількості їх значень. Для того, щоб масштабувати сигнал $u(n)$ в часі та отримати $v(n) = u(\alpha n)$, необхідно до $u(n)$ застосувати інтерполяцію порядку N, а потім децимацію порядку M так, щоб $\alpha = N/M$. На відміну від аналогових сигналів, які можна масштабувати в часі з будь-якими значеннями α, дискретні сигнали можуть бути масштабовані лише з раціональними α.

1.2.3 Частотний аналіз

В галузі обробки сигналів важливу роль грають різноманітні перетворення, які будують відображення оригінального сигналу в інші області, наприклад, в область частот. Це дозволяє спростити математичні операції для оцінювання поведінки системи, що породила сигнал. Одним з важливих спрощень є перехід від операції згортки в області часу до простого множення в області частот. Проте недоліком такого підходу є те, що після застосування всіх необхідних операції, перетворений сигнал має бути перетворено назад в область часу. До поширених перетворень належать фазорний аналіз (для синусоїд та періодичних сигналів), перетворення Фур'є, та перетворення Лапласа, проте найширше використовується саме перетворення Фур'є та його оптимізовані версії для дискретних сигналів.

Перетворення Фур'є дозволяє уніфікувати представлення як періодичних, так і неперіодичних функцій шляхом їх розкладання у ряд Фур'є. Зв'язок між періодичним сигналом $u_p(t)$ та його представленням у вигляді показникового ряду Фур'є виглядає наступним чином:

$$u_p(t) = \sum_{k=-\infty}^{\infty} U(k) e^{i2\pi k F_0 t},\ U(k) = \frac{1}{T}\int_{-T/2}^{T/2} u_p(t) e^{-i2\pi k F_0 t} dt. \tag{1.39}$$

Якщо період T періодичного сигналу $u_p(t)$ прямує до нескінченності, то сигнал перестає бути періодичним і натомість перетворюється на імпульс $x(t)$. Такий перехід від періодичного до неперіодичного сигналу також представляє перехід від потужнісного до енергетичного сигналу.

З $T \to \infty$ випливає $F_0 = 1/T \to 0$. У цьому випадку спектр значень коефіцієнтів ряду Фур'є стає неперервною кривою. Тобто, якщо замінити F_0 нескінченно малим значенням, то дискретні значення частот kF_0 в (1.39) можна замінити неперервним значенням частоти F. Це також означає, що множник інтеграла $1/T$ прямує до нуля, а отже і саме значення $U(k)$ прямує до нуля, що не дозволяє використовувати його для оцінки спектральних характеристик неперіодичного сигналу $u(t)$. Однак залежності значень $U(k)$ можна позбутися, якщо натомість розглядати значення $TU(k)$:

$$TU(k) \int_{-T/2}^{T/2} u_p(t) e^{-i2\pi k F_0 t} dt. \tag{1.40}$$

Не зважаючи на невизначеність $TU(k)$, права сторона рівняння (1.40) зазвичай існує для $T \to \infty$. А оскільки $kF_0 \to F$, цей інтеграл задає функцію від неперервної F, і якщо покласти $TU(k) = U(F)$ то отримуємо:

$$U(F) = \lim_{T \to \infty} TU(k) = \int_{-\infty}^{\infty} u(t) e^{-i2\pi F t} dt. \tag{1.41}$$

Рівняння (1.41) описує *перетворення Фур'є* $U(k)$ сигналу $u(t)$. Перетворення Фур'є дозволяє представити аперіодичні сигнали в області частот.

Періодичний сигнал $u_p(t)$ можна реконструювати з його спектральних коефіцієнтів $U(k)$:

$$u_p(t) = \sum_{k=-\infty}^{\infty} U(k) e^{i2\pi k F_0 t}. \tag{1.42}$$

Якщо $T \to \infty$, то розглядається неперіодичний сигнал $x(t)$, а замість $U(k)$ розглядається величина $TU(k)$. Для цього домножимо і поділимо праву частину (1.42) на T та замінимо $1/T$ на F_0:

$$u_p(t) = \sum_{k=-\infty}^{\infty} TU(k) e^{i2\pi k F_0 t} T^{-1} = \sum_{k=-\infty}^{\infty} TU(k) e^{i2\pi k F_0 t} F_0. \tag{1.43}$$

Дискретні частоти kF_0 перетворюються на неперервну частоту F, а оператор суми замінюється інтегруванням:

$$x(t) = \int_{-\infty}^{\infty} U(f) e^{i2\pi Ft} df. \tag{1.44}$$

Рівняння (1.44) називається *оберненим перетворенням Фур'є* та дозволяє отримувати сигнал $u(t)$ з його спектра $U(f)$.

З неперервних форм прямого (1.41) та оберненого (1.44) перетворень Фур'є, що визначають зв'язок між неперервним сигналом $u(t)$ в області часу та його неперервним спектром $U(f)$ в області частот, завжди можна отримати їх дискретну форму, що описувала б періодичний сигнал $u_p(t)$ з періодом $T = 1/F_0$. Формально, дискретизація в області частот відповідає періодичному розширенню в області часу. Крок дискретизації ΔF в області частот обернений до періоду реконструйованого сигналу $u_p(t)$. Якщо $u(t)$ обмежений в часі з $t_0 < T$, його періодичне розширення полягає у простому періодичному продовженню $u(t)$. Ці властивості перетворення Фур'є встановлюють важливий зв'язок між областями часу та частот будь-якого сигналу.

Перетворення Фур'є, взагалі кажучи, комплексне та може бути представлене в наступних еквівалентних формах:

$$\begin{aligned} U(f) &= \Re\{X(f)\} + i\Im\{U(f)\} \\ &= |U(f)|\hat{\phi}(f) \\ &= |U(f)|e^{i\phi(f)}. \end{aligned} \tag{1.45}$$

Для дійсних сигналів $U(f)$ спряжено симетричне та $U(-f) = U^*(f)$. Це означає, що величини $|U(f)|$ або $\Re\{U(f)\}$ парно симетричні, а фази $\phi(f)$ або $\Im\{U(f)\}$ непарно симетричні.

Перетворення Фур'є $U(f)$ дійсного парно симетричного сигналу $u(t)$ завжди дійсна та парно симетрична функція від f та записується у формі $U(f) = A(f)$. Перетворення Фур'є $U(f)$ дійсного непарно симетричного сигналу $u(t)$ завжди уявна та непарно симетрична відносно f та записується у формі $U(f) = iA(f)$. Для таких сигналів, зручно користуватися лише спектром амплітуд $A(f)$.

Означення 1.6. *Спектр (частотна густина) сигналу $u(t)$ – це перетворення Фур'є його автокореляційної функції $r_{uu}(t)$.*

Перетворення Фур'є дискретного часу (не плутати з дискретним перетворенням Фур'є) описує спектр дискретних сигналів та формалізує поняття їх періодичного спектра. Ідеальне семплування аналогового сигналу $u(t)$ дає ідеально дискретизований сигнал $u_I(t)$, що має періодичний спектр $U_p(f)$:

$$\begin{aligned} u_I(t) &= \sum_{k=-\infty}^{\infty} u(k\Delta t)\delta(t - k\Delta t) \\ U_p(f) &= S \sum_{k=-\infty}^{\infty} X(f - kS). \end{aligned} \tag{1.46}$$

Якщо ввести заміну $\delta(t - \alpha) \Leftrightarrow exp^{-i2\pi\alpha f}$, то спектр $U_p(f)$ можна записати як:

$$U_p(f) = \sum_{k=-\infty}^{\infty} x(k\Delta t)e^{-i2\pi k\Delta t f}, \tag{1.47}$$

де $U_p(f)$ – періодичний спектр з періодом S. Щоб відновити аналоговий сигнал $u(t)$, необхідно застосувати фільтр низьких частот до дискретизованого сигналу з передавальним коефіцієнтом $1/S$ для частот в проміжку $-S/2 \leq f \leq S/2$. Виконавши це, отримаємо обернену операцію до дискретного перетворення Фур'є:

$$\begin{aligned} u(t) &= \frac{1}{S}\int_{-S/2}^{S/2} U_p(f)e^{i2\pi ft}df \\ u(n\Delta t) &= \frac{1}{S}\int_{-S/2}^{S/2} U_p(f)e^{i2\pi fn\Delta t}df. \end{aligned} \tag{1.48}$$

Рівняння (1.47) і (1.48) задають пару трансформувань, що дозволяють отримувати періодичний спектр $U_p(f)$ ідеально дискретизованого сигналу з його семплів $u(n\Delta t)$, а також відновлювати їх з отриманого спектра. Ці операції є аналогами перетворення Фур'є для неперервних сигналів та їх дискретних спектрів $U(k)$. Якщо використати *цифрову частоту* $F = f/S$ та позначити $u(n\Delta t) = u(k)$, отримаємо:

$$U_p(F) = \sum_{k=-\infty}^{\infty} u(k)e^{-i2\pi kF}, \tag{1.49}$$

$$u(k) = \int_{-1/2}^{1/2} U_p(F)e^{i2\pi nF}dF. \tag{1.50}$$

На практиці для роботи з сигналами зазвичай використовується N-точкове дискретне перетворення Фур'є (discrete Fourier transform, DFT) та обернене дискретне перетворення Фур'є (inverse discrete Fourier transform, IDFT). Задаються ці перетворення наступним чином:

$$U_{\mathrm{DFT}}(k) = \sum_{n=0}^{N-1} u(n) e^{-i2\pi nk/N}, \tag{1.51}$$

$$u(n) = \frac{1}{N} \sum_{k=0}^{N-1} U_{\mathrm{DFT}}(k) e^{i2\pi nk/N}. \tag{1.52}$$

Кожен з цих виразів складається з N рівнянь. Кожен семпл DFT – це зважена сума усіх семплів сигналу $u(n)$. Однією з найважливіших властивостей DFT та IDFT є періодичність. Експонента $e^{\pm i2\pi k/N}$, що входить в обидва рівняння, періодична як відносно n, так і відносно k, з періодом N:

$$e^{i2\pi k/N} = e^{i2\pi(n+N)k/N} = e^{i2\pi n(k+N)/N}$$

Через це DFT та IDFT періодичні з періодом N і для роботи з ними достатньо обчислити лише один період (від 0 до $N-1$). Традиційно, нумерація $u(n)$ та $U_{\mathrm{DFT}}(k)$ починається з 0. Константу N називають *розширенням* дискретного перетворення Фур'є, оскільки вона обмежує максимальну кількість складових частот сигналу, обраних для аналізу.

1.2.4 Спектрограма

Спектр сигналу або його спектральна густина (power spectral density, PSD – це перетворення Фур'є його автокореляційної функції $r_{uu}(t)$, яка є дійсною, ненегативною, парною функцією зі значенням $R_{uu}(0)$, що рівне середній потужності сигналу $u(t)$. Якщо сигнал $u(t)$ було дискретизовано та для дослідження доступна лише невелика частка семплів, то замість використання автокореляційної функції вдаються до методів приблизної оцінки значень PSD з наявних даних. Причина цьому – схильність значень спектра скінченних послідовностей до спотворень типу *спектрального витоку* та низького розширення перетворення Фур'є. Виділяють два підходи до оцінки значень спектра – *непараметричний* та *параметричний*. Непараметричний метод не робить жодних припущень що-

до природи аналізованих даних. Параметричний метод моделює дані як відгук цифрового фільтра на випадковий сигнал з постійною потужністю. Обидва методи покладаються на статистичні міри для оцінки якості обчисленого PSD.

Найпростіший непараметричний метод оцінки *періодограми* $P(k)$ заснований на обчисленні DFT для N семплів дискретного сигналу $u(n)$:

$$P(k) = \frac{1}{N}|U_{\mathrm{DFT}}(k)|^2. \tag{1.53}$$

Цей метод дає прийнятні оцінки спектральної густини для детермінованих сигналів з обмеженим діапазоном частот, що дискретизовані з частотою, більшою за частоту Найквіста (див. визначення 1.5). Проте для сигналів, у яких присутній шум, якість цього метода падає та не може бути покращена збільшенням кількості семплів N, хоч роздільна здатність DFT і зросте.

Метод Велча для оцінки спектральної густини, що широко використовується у практичних задачах, заснований на усередненні значень періодограм, що перетинаються. Оригінальний сигнал $u(n)$ довжиною N розбивається на K сегментів довжиною M, що перетинаються з довжиною перетину, що становить D семплів. Зазвичай обирають довжину перетину $D = M/2$ або $D = 3M/4$. Для кожного сегмента застосовується *вікно* $w(n)$ довжиною M, щоб запобігти спектральним спотворенням. Після цього для кожного сегмента обчислюється його періодограма (1.53). Отримані значення періодограм довжиною M усереднюються для отримання фінальної оцінки періодограми фрагмента сигналу $u(n)$ довжиною N. Зазвичай отриману періодограму також нормалізують, тобто ділять на потужність сигналу, що задає значення вікна ($\sum |W_{\mathrm{DFT}}(k)|^2$ або $\frac{1}{M}\sum |w(n)|^2$). При збільшенні довжини сегментів M зростає роздільна здатність періодограми, а при збільшенні їх кількості K – її гладкість (хоча при цьому є ризик втрати дрібних деталей). Баланс значень M і K має обиратися дослідниками залежно від конкретної задачі.

У багатьох задачах також важливо знати як спектральні характеристики сигналу змінюються з часом. Простий підхід до отримання такої інформації полягає у послідовному застосуванні методів оцінки періодограми для послідовності сегментів досліджуваного сигналу, що перетинаються. Потім періодограми, отримані для кожного сегмента, один за одним наносяться на графік, який називають *часово-частотним*. Цей графік являє собою теплокарту, гори-

зонтальною віссю якої є час, вертикальною – значення k-ї комірки періодограми, а частотна густина в k-й комірці в час t кодується кольором.

Головним обмеженням при отриманні часово-частотної інформації, особливо для коротких сигналів, це неможливість одночасної локалізації часу та частоти з однаковою точністю. Для отримання інформації про низькі частоти, довжина сегмента M має бути досить довга для вміщення їх періодів, але таке збільшення довжини призводить до втрати інформації про поведінку високих частот на таких довгих періодах. Проте існують методи, які дозволяють отримати кращу часову роздільну здатність у високих частотах і при цьому не втрачати здатність отримувати інформацію про низькі частоти. Зазвичай такі методи засновані на використанні *вейвлетів*, які, аналогічно до розкладання сигналу в ряд Фур'є, використовують альтернативні функції для формування зваженої суми.

1.2.5 Фільтрування

Фільтри є одним з найважливіших інструментів для аналізу та обробки сигналів. Будь-яка система для цифрової обробки сигналів містить мінімум один фільтр низьких частот, який обмежує діапазон частот вхідного сигналу для уникнення спотворень типу аліасингу. Більшість задач обробки сигналів можна звести саме до побудови фільтрів.

Означення 1.7. *Фільтр – інструмент для вибору частот зі спектра сигналу та маніпулювання їх амплітудами.*

Термінологія, пов'язана з фільтрами, заснована на описі змін компонентів спектра досліджуваного сигналу, та описує *частотний відгук* фільтра $H(f)$ після його застосування на деякому сигналі. Частотний відгук фільтра можна оцінити, якщо застосувати його на сигналі, що містить у собі усі частоти в рівних долях, тобто його спектр виглядає як горизонтальна пряма лінія. Перетворення Фур'є відфільтрованого сигналу буде задавати частотний відгук фільтра, а його спектр $|H(f)|$ – передавальний коефіцієнт фільтра.

Фільтри вибору частот – це інструменти, що пропускають частоти з певного діапазону і блокує решту частот. Діапазон частот, що пропускається (затримуються), називається *смугою пропускання (затримання)*. Границі смуг фільтра називається *частотами зрізу*. Ідеальний фільтр – це функція перенесення, що

зберігає величини усіх частот у смузі пропускання, не пропускає жодні частоти зі смуги затримання, та не вносить жодних фазових змін у вихідний сигнал. Результатом застосування фільтра із постійним передавальним коефіцієнтом $|H(f)|$ у смузі пропускання є масштабування амплітуди вхідного сигналу $u(t)$. Якщо фільтр пропускає усі компоненти спектра з однаковим запізненням, то вихідний сигнал має *лінійний фазовий зсув*. Якщо значення $|H(f)|$ у смузі пропускання змінюються, виникають *спотворення амплітуди*. Якщо різні компоненти спектра вихідного сигналу мають різні запізнення, то виникають *фазові спотворення*. Фільтр, який не має ні амплітудних, ні фазових спотворень, називають *ідеальним*.

До цифрових сигналів застосовуються цифрові фільтри.

Цифрові фільтри зазвичай описуються різницевими рівняннями у наступній формі:

$$y(n)+A_1y(n-1)+\cdots+A_Ny(n-N)=B_0x(n)+B_1x(n-1)+\cdots+B_Mx(n-M). \tag{1.54}$$

Рівняння (1.54) – *рекурсивний* фільтр, поточний вивід якого залежить від його попередніх значень $y(n-k)$, а також поточного вхідного значення. Такі фільтри також називають фільтрами з нескінченними імпульсними характеристиками (infinite impulse response, IIR), оскільки він може мати потенційно нескінченний відгук (ненульові значення, згенеровані фільтром для послідовності входів $\{1,0,0,\dots\}$). Константа N називається *порядком* IIR фільтру.

Розглянемо тепер наступне різницеве рівняння:

$$y(n)=B_0x(n)+B_1x(n-1)+\cdots+B_M(n-M). \tag{1.55}$$

Його поточний відгук залежить лише від поточного входу та не містить у своєму визначенні. попередніх відгуків. Фільтри, що задаються рівняннями типу (1.55) називаються *нерекурсивними* або фільтрами ковзного середнього, оскільки його відгук задається зваженою сумою (ковзним середнім) вхідних значень. Такі фільтри також називають фільтрами зі скінченними імпульсними характеристиками (finite impulse response, FIR), оскільки їх відгук завжди обмежений.

1.2.6 Кепстральний аналіз

Слово “кепстр” отримується зі слова “спектр” шляхом перевертання перших чотирьох літер. Неформально кепстр можна описати як “спектр спектра”, що доволі точно передає принцип отримання кепстра даного сигналу – спектр описує періодику, яка присутня у сигналі, тоді як кепстр описує періодику, присутню в самому спектрі. Іншими словами, кепстр використовується для виявлення періодичностей між піками та впадинами в спектрі. Для прикладу, наведемо кепстр для сигналу з Рис. 1.7 його спектру та кепстру, наведено на Рис. 1.5

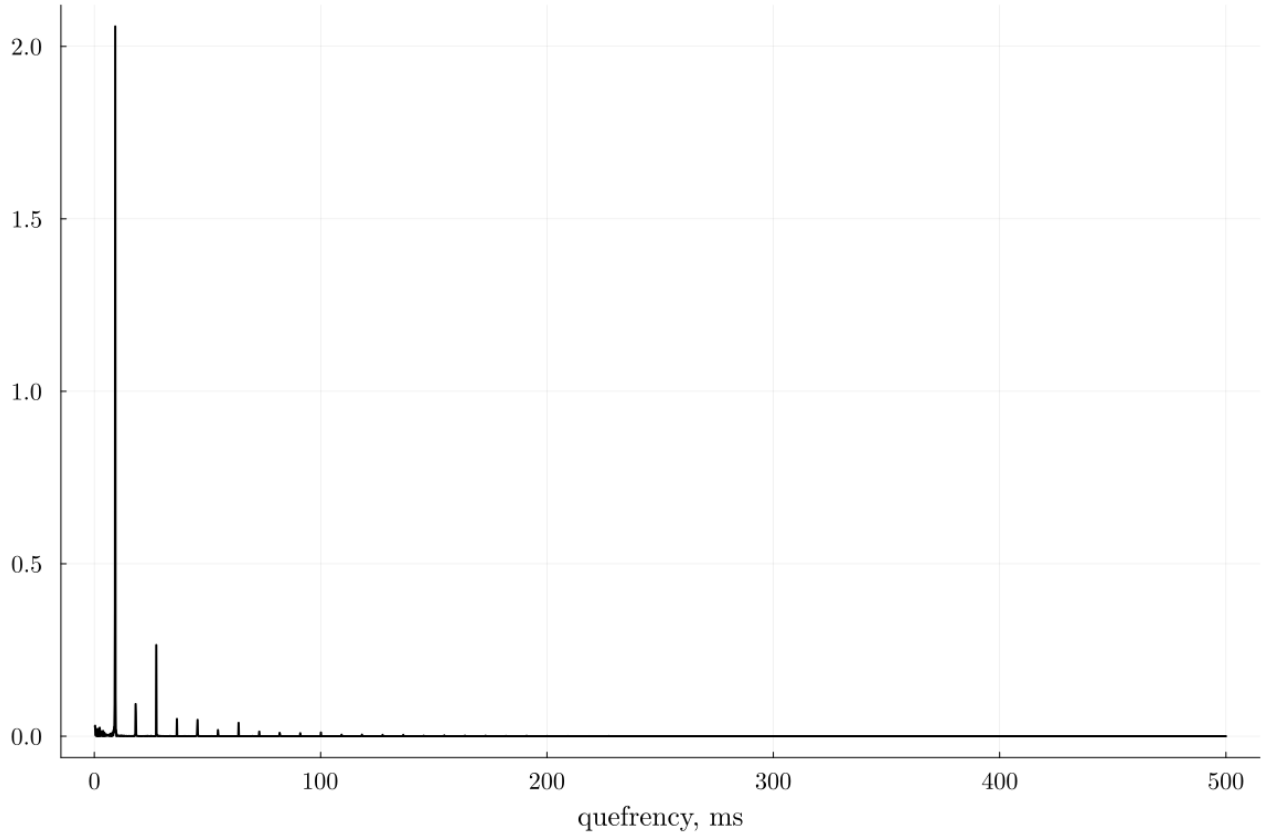


Рис. 1.5: Приклад кепстру гармонічного сигналу з Рис. 1.7. Перший елемент кепстру було відкинуто, оскільки він відповідає нульовій частоті спектра, тобто наскільки він відхиляється від осі X. Перший пік на графіку відповідає найбільш високочастотній періодичності, яка присутня у спектрі – гармонікам фундаментальної частоти 110 Гц, що відповідає кепстральній частоті ~ 13 мілісекунд.

Існує декілька видів кепстрів: комплексний, дійсний, густинний, фазовий і т.д. На практиці зазвичай користуються густинним кепстром (Power Cepstrum), визначення якого наведено в Означенні 1.8 [45].

Означення 1.8. *Кепстр (густинний) – це квадрат оберненого перетворення Фур'є логарифмів квадратів значень перетворення Фур'є сигналу:*

$$c(n) = \left| U_{IDFT}\left(\log\left(|U_{DFT}(u(t))|^2\right)\right)\right|^2. \tag{1.56}$$

Значення кепстру існують в особливій часовій області, значення часу $c_{quef}(n)$ у якій називають кепстральним (quefrency).

Кепстральний аналіз дозволяє оцінити "ступінь гармонічності" досліджуваного сигналу та фундаментальні частоти гармонічних коливань, що у ньому присутні. Оскільки кепстр обчислюється як обернене перетворення Фур'є, то його значення знаходяться в часовій області, а не в частотній. Практично це означає, що компоненти кепстра позначають густину *часових періодів* в спектрі. Наприклад, на Рис. 1.5 пік в кепстрі на позначці 13 мілісекунд означає, що у кепстрі присутня сильна періодичність, зумовлена частотою $f_s \cdot c_{\text{quef}}(i) = 8000 \cdot 0.014 = 112$ Гц та її гармоніками (точнішу оцінку фундаментальної частоти можна отримати, якщо підвищити роздільну здатність дискретного перетворення Фур'є або дискретизувати сигнал з вищою частотою).

Кепстральний аналіз широко застосовується у різноманітних галузях для отримання ознак досліджуваного сигналу. Наприклад, кепстр та його різновиди використовують для ідентифікації мовця [46], визначення джерел сейсмічної активності [47], та у багатьох прикладних задачах механіки, радіо, та електроніки [48].

1.3 Голосовий сигнал як об'єкт дослідження

Як згадувалось раніше, голосовий сигнал та робота з ним є популярною темою досліджень, що часто спрямовані на розв'язання прикладних задач. Однією з найважливіших вимог, що є частиною таких задач, є необхідність роботи розроблених методів та алгоритмів в реальному часі. Наприклад, система TTS має перетворювати текст, наданий користувачем, у мовлення з мінімальною затримкою, система SI має швидко розпізнавати мовця, система VC має перетворювати один сигнал в інший з мінімальним відставанням. Саме це і спонукає дослідників продовжувати пошуки моделей, які за своєю точністю та якістю результатів не поступаються існуючим, але водночас є більш економічними та

швидкими в застосуванні.

Методи, пов'язані з обробкою мовлення, зазвичай тісно пов'язані із фундаментальною моделлю, що використовується для представлення та параметризації джерела голосового сигналу. Тобто можна сказати, що модель, яка здатна якісно синтезувати голос, є базовою для усіх інших моделей, які або параметризують її для досягнення бажаного вихідного сигналу (наприклад, моделі TTS і VC), або інтерпретують реальний голосовий сигнал з точки зору цієї моделі з метою ідентифікації та інтерпретації її параметрів (наприклад, в моделях SI та STT).

Фундаментальні моделі можна поділити на три основних типи: акустичні, артикуляторні, та моделі на основі кодування природного мовлення [49]. Акустична модель полягає у побудові співвідношень між акустичною інформацією фрагментів мовлення та мовними конструктами. Артикуляторна модель має на меті побудову фізичної симуляції голосового апарату людини для симуляції проходження через нього повітря. Головна ідея моделей на основі кодування полягає у тому, що природній голосовий сигнал можна закодувати та представити таким чином, щоб із такого представлення можна було синтезувати нові голосові сигнали. Зазвичай кодування виконується як параметризація моделі джерела-фільтра – параметри, що задають тип та поведінку джерела, визначають фундаментальну частоту сегмента мовлення, що містить голосні звуки, або рівень шуму для моделювання приголосних, а параметри фільтра задають його резонансні характеристики та визначають тип звуку, що вимовляється.

Для практичних застосувань, що передбачають обробку та перетворення голосових сигналів, часто використовується саме кодувальна модель природного мовлення на основі джерела-фільтра, що здебільшого зумовлено цифровим представленням голосових сигналів в комп'ютерах та суміжних пристроях та легкістю представлення параметрів моделі у цифровому вигляді. Тому надалі під моделлю голосового сигналу будемо розуміти саме такі моделі.

Для ефективного моделювання голосового сигналу у вигляді комбінації джерела і фільтру, необхідно розуміти будову голосового тракту людини, функції його складових частин та поведінку повітряного потоку за їх різної конфігурації.

Голос – це сукупність різних за частотою, інтенсивністю та тембром звуків, що вимовляються людиною за допомогою голосового апарату. Голосовий апа-

рат – це комплекс органів і систем, які беруть участь у голосоутворенні. Легені відіграють роль джерела енергії. Повітря з легень під тиском проходить через трахею і голосові зв'язки. Голосові зв'язки замикаються й розмикаються, модулюючи повітряний потік, внаслідок чого він набуває вигляду послідовності імпульсів складної форми. Схематичне зображення голосового апарату людини наведено на Рис. 1.6.

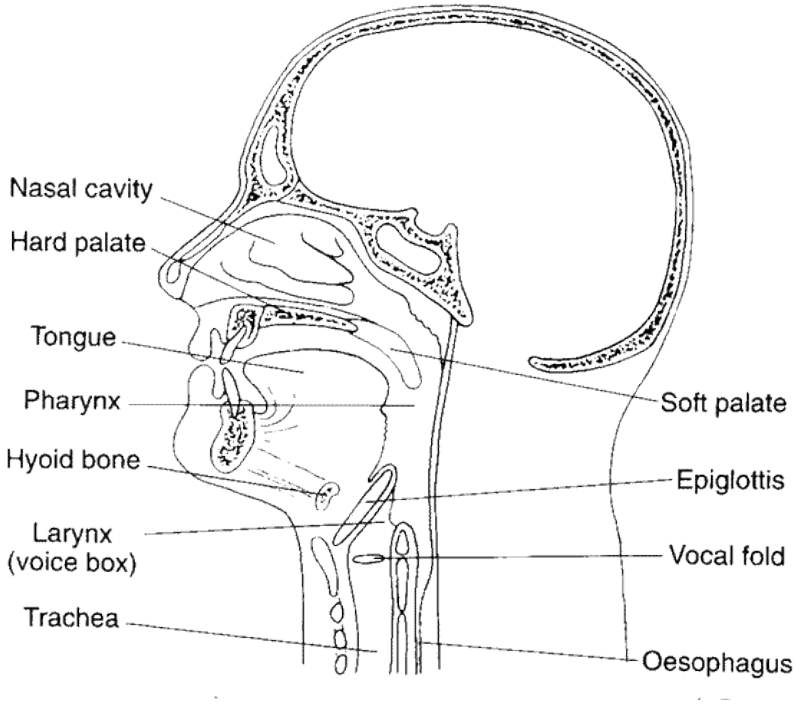


Рис. 1.6: Схема голосового апарату людини [50].

Деякі індивідуальні відмінності в будові голосових зв'язок обумовлюють відмінності у формі імпульсів, а значить, тембру голосу. Частота імпульсів називається частотою основного тону та позначається F_0. Вона грає головну роль в утворенні інтонацій і співі, а також є опорною для усіх голосних звуків. Складність форми імпульсів полягає у присутності в них гармонік основного тону, що мають різну інтенсивність. Приклад гармонічного сигналу та його спектру наведено на Рис. 1.7.

Означення 1.9. *Фундаментальна частота (F_0) – це найнижча частота періодичної хвилі.*

Через гортань повітря потрапляє в ротову і носову резонансні порожнини, що розділені піднебінною завіскою, та розділяється між ними на два потоки в

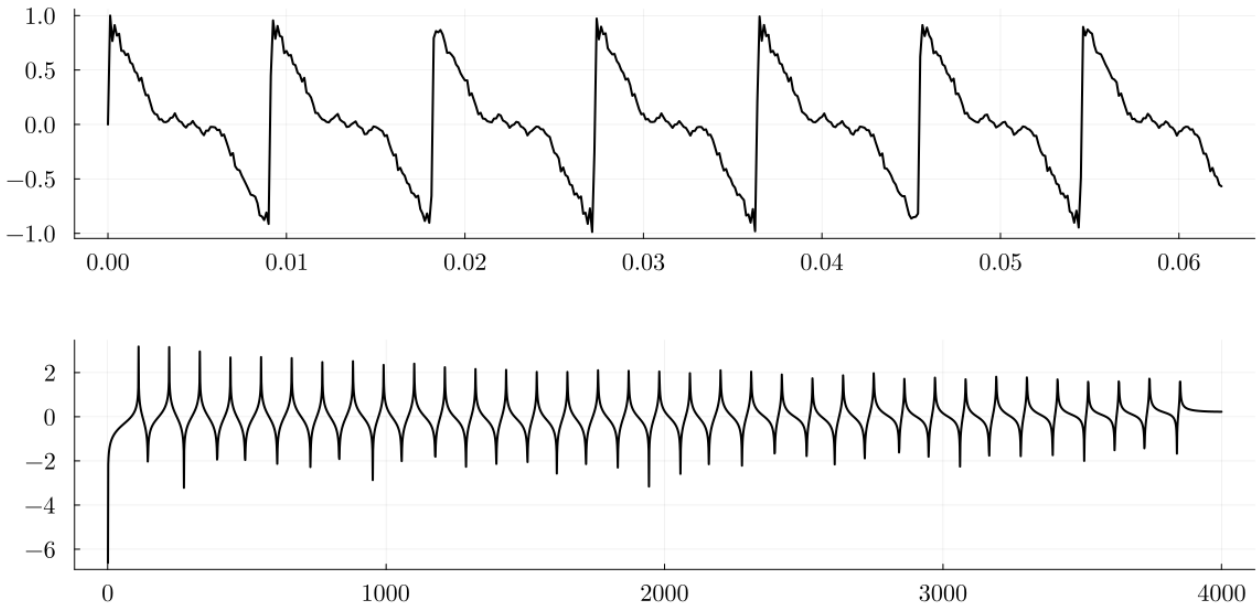


Рис. 1.7: Приклад гармонічного звукового сигналу з основною частотою 110 Гц (зверху), що може породжуватися гортанню при проходженні через неї повітря, та його спектра (знизу). Частота семплування – 8 кГц.

різних пропорціях. При значному перерозподілі в носову порожнину звуки назалізуються, тобто набувають носового тембру. За відповідної артикуляції язика, зубів і губ у голосовому тракті можуть утворюватися щілини. При проходженні через них у потоці повітря виникає високочастотна турбулентність, що на слух сприймається як шум. Залежно від артикуляції, цей шум може фільтруватися та утворювати деякі типи приголосних [51].

У процесі мовлення форми ротової та носової резонансних порожнин змінюються через рух язика, нижньої щелепи, губ, відкриття і розкриття голосових зв'язок. Внаслідок цього насичений гармоніками звуковий сигнал, породжений гортанню, фільтрується. Гармоніки, частота яких найближча до поточних резонансних частот обох порожнин, підсилюються, а решта – пригнічується. Значення частот, в околі яких гармоніки підсилюються, називаються *формантними*, а загальна спектральна характеристика відфільтрованого звукового сигналу – *тембром*. Кількість та відносна інтенсивність формант визначає тип звуку, що вимовляється. Наприклад, за деякими даними, для **a** - це приблизно 700 і 1200 Гц, для **o** – 400 і 800 Гц, для **y** – 300 і 700 Гц, для **i** – 200 і 2200 Гц, і так далі.

Означення 1.10. *Комбінація якостей звуку, які відрізняють його від інших звуків з однаковою частотою та інтенсивністю, називається тембром.*

Нарешті, коли повітряний потік залишає голосовий апарат, він спричиняє хвильові зміни тиску в навколишньому середовищі. Величина, що описує амплітуду таких коливань, називається інтенсивністю або гучністю звуку.

Означення 1.11. *Гучність звуку – відносна величина, що вимірюється в фонах і чисельно дорівнює рівню звукового тиску (в децибелах – дБ), створюваного синусоїдальним тоном з частотою 1 кГц такої ж гучності.*

З наведених вище фізіологічних характеристик голосового тракту маємо три визначні характеристики будь-якого голосу: висоту основного тону F_0, тембр, і гучність.

Як зазначалося вище, модель мовлення типу джерело-фільтр можна параметризувати за допомогою деяких визначних характеристик, що описують поточний стан голосового тракту мовця. До цих визначних характеристик належать як векторні, так і скалярні величини. До скалярних належать фундаментальна частота та гучність звуку, а тембр є вектором, що описує інтенсивність частотних компонент сигналу.

Проте мовлення – динамічний процес, характеристики якого змінюються з часом. Через це голосовий сигнал розбивають на невеликі сегменти (вікна), для кожного з яких характеристики обчислюють окремо. Таким чином отримується послідовність значень, яка описує стан голосового тракту мовця у певні моменти часу. Приклад віконного підходу до аналізу мовлення наведено на Рис. 1.8

Існують різні підходи до розбиття сигналу на сегменти. Усі вони направлені на покращення якості аналізу та недопущенню можливих спотворень сигналу після розбиття.

1.4 Висновки

У розділі розглянуто мовлення як сигнал, який можна аналізувати та синтезувати за допомогою інструментів, розроблених в рамках різних математичних та інженерних дисциплін. Кожен з цих інструментів детально розглянуто, введено необхідну для їх опису термінологію, а також описано зв'язки між ними. Введено поняття динамічної системи та обґрунтовано доцільність моделювання голосового сигналу саме як динамічного процесу. Розглянуто різні підходи до моделювання, наведено їх переваги та недоліки.

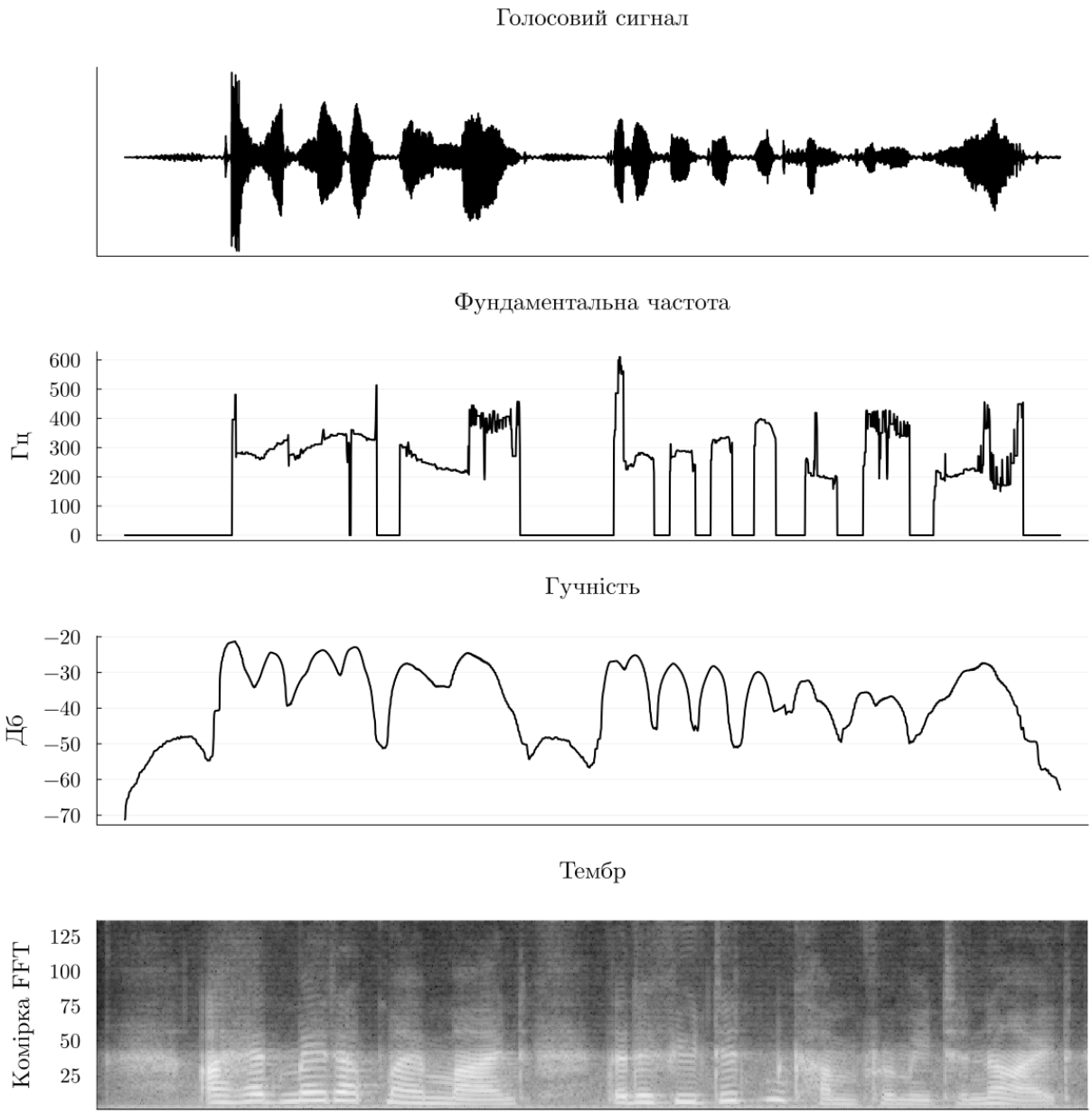


Рис. 1.8: Приклад виділення визначних характеристик голосового сигналу шляхом його розбиття на сегменти, що перетинаються, та застосування методів для оцінки зміни його фундаментальної частоти, гучності, та тембру у часі

З викладеного матеріалу маємо, що дослідження можливості побудови моделі, яка поєднує в собі гнучкість нейронних мереж та пояснювальну силу класичного підходу до побудови моделей, є доцільним та вартим подальшого розвитку.

Розділ 2

Нова модель для розв'язання задачі керування системою ЗДР у латентному просторі

Нейронна система звичайних диференціальних рівнянь (neural ODE, NODE) – це новітнє сімейство залишкових нейромереж, яке замість того, щоб явно визначати дискретну послідовність прихованих шарів, параметризує похідну свого внутрішнього стану за допомогою диференціального рівняння, права частина якого задана нейромережею [43]. Оригінальна стаття пропонує NODE як альтернативу залишковим нейронним мережам, що виконують серію перетворень свого прихованого стану:

$$\mathbf{h}_{t+1} = \mathbf{h}_t + f(\mathbf{h}_t, \Theta), \tag{2.1}$$

де $t \in [0..T]$ і $\mathbf{h} \in \mathbf{R}^D$. Розглядаючи рівняння (2.1) як формулу чисельного інтегрування методом Ейлера, автори роблять перехід від дискретного до неперервного представлення оригінальної послідовності:

$$\frac{d\mathbf{h}(t)}{dt} = f(\mathbf{h}, t, \Theta), \tag{2.2}$$

таким чином отримуючи залишкову нейромережу, яка потенційно є нескінченно глибокою. Результат такої нейромережі обчислюється як результат інтегруван-

ня її правої частини:

$$F(x_0, T) = x_0 + \int_0^T f(x, \Theta, t)\, dt, \tag{2.3}$$

де x_0 – початкові умови, а Θ – параметри нейромережі.

Такий підхід можна застосовувати не тільки для залишкових мереж. Ефективний метод для зворотного поширення помилок за допомогою оберненого автоматичного диференціювання, розробленого авторами NODE, дозволяє створювати та тренувати будь-які моделі, що містять NODE як один з компонентів.

Одним з недоліків базової NODE моделі є те, що вона здатна вивчити лише один неперервний потік F. Іншими словами, набір параметрів Θ є статичним і не може адаптуватися до нових вхідних даних або змінюватися з часом. Це доволі сильно обмежує коло застосування NODE, особливо коли необхідно розробити модель для рядів, які змінюють свої якісні характеристики з плином часу. Ще одним недоліком NODE є те, що поведінка (2.3) здебільшого визначається розмірністю простору вхідних даних, оскільки структура моделі вимагає, щоб функція F була гомеоморфізмом простору вхідних даних самого в себе, що особливо ускладнює її застосування для моделювання одновимірних часових рядів.

Для усунення цих обмежень та розширення можливостей моделей на базі NODE було запропоновано декілька різних підходів.

Одна з таких моделей – Процес Нейронної Системи ЗДР (Neural ODE Process. NODEP) [52]. Вона досягає адаптивності моделі до поточних даних шляхом застосування стохастичних методів для визначення та підтримування найбільш ймовірних значень набору параметрів Θ. Сама модель полягає у тому, щоб здійснити перехід до латентного представлення вхідних даних за допомогою нейромережі-кодувальника, отримати набір параметрів із наданого контексту, знайти траєкторію вкладеної NODE у латентному просторі з використанням отриманих параметрів, та декодувати кожну з точок отриманої траєкторії нейромережею-декодувальником для отримання траєкторії у просторі вхідних даних.

Ще одним прикладом розширення базової NODE є нейронно-контрольована система ЗДР (neurally-controlled ODE, N-CODE) [53]. На відміну від NODEP,

вона реалізує детермінований підхід до визначення значень Θ, який полягає у їх безпосередньому включенні до самої системи у якості змінних. Таким чином вхідні дані зливаються зі значеннями стану у єдину систему та інтегруються пліч-о-пліч, що значно підвищує розмірність системи та розширює спектр її можливих поведінок та способів представлення прихованої інформації про дані.

Обидві моделі NODEP та N-CODE – це приклади нейронних гіпермереж, оскільки вони виводять свої параметри не на етапі навчання, а безпосередньо під час роботи залежно від оброблюваних на цей момент даних. Проте підхід, що використовує NODEP, більш гнучкий, оскільки він не сполучає вхідні дані та параметри в єдиний простір. Це дозволяє проєктувальникам моделі задавати бажану розмірність латентного простору для досягнення бажаної поведінки траєкторій, а також вводити обмеження на можливі значення параметрів, щоб зменшити ймовірність виникнення небажаних поведінок.

Нова модель, нейронна гіперсистема ЗДР (hyper NODE, hNODE), пропонує дещо спрощений та детермінований підхід до моделювання, запропонований в NODEP, з використанням підходу до контролю, що використовується в N-CODE.

Розглянемо скалярний часовий ряд $U = (u_0, u_1, \ldots, u_n)$ з часовим кроком Δt. Мета аналізу часових рядів – вилучення корисної та значущої інформації з U та передбачення його майбутніх значень. Для досягнення цієї мети, можна побудувати модель F, яка відображає множину U саму в себе:

$$U^* = F(U, \Theta), \tag{2.4}$$

де $U^* = (u_{n+1}, u_{n+2}, \ldots, un + m)$ – прогнозовані значення часового ряду.

Модель F, що керується набором параметрів Θ, має інтерпретувати послідовність U та обчислити передбачення для її наступних значень U^*. Проте якщо модельований процес, який породжує значення часового ряду, змінюється, модель F стає непридатною до використання, оскільки вона не може адаптувати свої параметри до нових умов та продовжувати надавати надійні результати. Для прикладу, розглянемо просту лінійну модель

$$F(u, A, b) = Au + b, \tag{2.5}$$

яка бере певну кількість спостережень з U та повертає передбачення U^*. Па-

раметри такої моделі статичні, а їх значення отримуються шляхом розв'язання задачі оптимізації, що мінімізує похибку моделі на тренувальних даних. В таку модель можна ввести функції контролю $A'(u)$ та $b'(u)$ з параметрами θ_A і θ_b, щоб дати моделі змогу корегувати власні параметри залежно від вхідних даних:

$$\begin{aligned} F^*(u, A, b, \theta_A, \theta_b) &= (A + A'(u))u + (b + b'(u)) \\ &= \mathcal{A}(u)u + \mathcal{B}(u). \end{aligned} \tag{2.6}$$

Вирази $\mathcal{A}$ і $\mathcal{B}$ – це відповідно матрична та векторна функції, які задають правила з отримання значень параметрів для рівняння (2.5) залежно від вхідних параметрів та набору гіперпараметрів, які визначають поведінку функцій $A'(u)$ і $b'(u)$. Якщо ми об'єднаємо гіперпараметри у єдину множину $\Theta = \{\theta_A, \theta_b\}$ та позначимо $\mathcal{H}(u, \Theta)$ як відображення з Декартового добутку набору гіперпараметрів Θ і вхідних значень X в множину $\Theta^* = \{A^*, b^*\}$ скорегованих параметрів для (2.5), то рівняння (2.6) набуде вигляду:

$$F^*(u, A, b, \Theta) = F(u, \mathcal{H}(u, \Theta)). \tag{2.7}$$

Якщо замість рівняння (2.5) підставити модель (2.3), то розмірність входів U може стати проблемою. Щоб зменшити або збільшити її, можна використати пару векторних функцій: одну для кодування, а другу – для декодування значень U:

$$\begin{aligned} V &= \mathcal{E}(U, \theta_\mathcal{E}) \\ U &= \mathcal{D}(V, \theta_\mathcal{D}). \end{aligned} \tag{2.8}$$

Тоді модель (2.7) можна записати у вигляді:

$$F^*(u, A, b, \Theta) = \mathcal{D} \circ F(\mathcal{E}(u, \Theta), \mathcal{H}(u, \Theta)). \tag{2.9}$$

Нарешті, модель hNODE визначається як (2.9), у якій функцією F виступає модель NODE (2.3):

$$hNODE(U, \Theta) = \mathcal{D} \circ G(\mathcal{E}(U, \Theta), \mathcal{H}(U, \Theta)), \tag{2.10}$$

де $\mathcal{E} : U \times \Theta \to L$ – кодувальник, що відображає дані часового ряду U в латентний простір L, $\mathcal{D} : L \to U$ – декодувальник, який інтерпретує точки з L і відображає їх в простір часового ряду U, $G : L \times \Theta^* \to L$ – система звичайних

диференціальних рівнянь, що задає закон еволюції стану моделі в латентному просторі, $\mathcal{H} : U \times \Theta \to \Theta^*$ – функція, яка задає правило визначення контролю для G. Схематичне зображення моделі (2.10) наведено на Рис. 2.1.

Означення 2.1. *Модель $\mathcal{E}$, що перетворює вхідні дані та будує відображення часового ряду на деякий латентний простір називається* ***кодувальником****. Модель $\mathcal{D}$, яка інтерпретує точки в латентному просторі та будує відображення з латентного простору в простір вхідних даних, називається* ***декодувальником****. Пара моделей $\mathcal{E}(U)$ та $\mathcal{D}(V)$, що задовольняють умові $U \approx \mathcal{D} \circ \mathcal{E}(U)$, називається* ***автокодувальником****.*

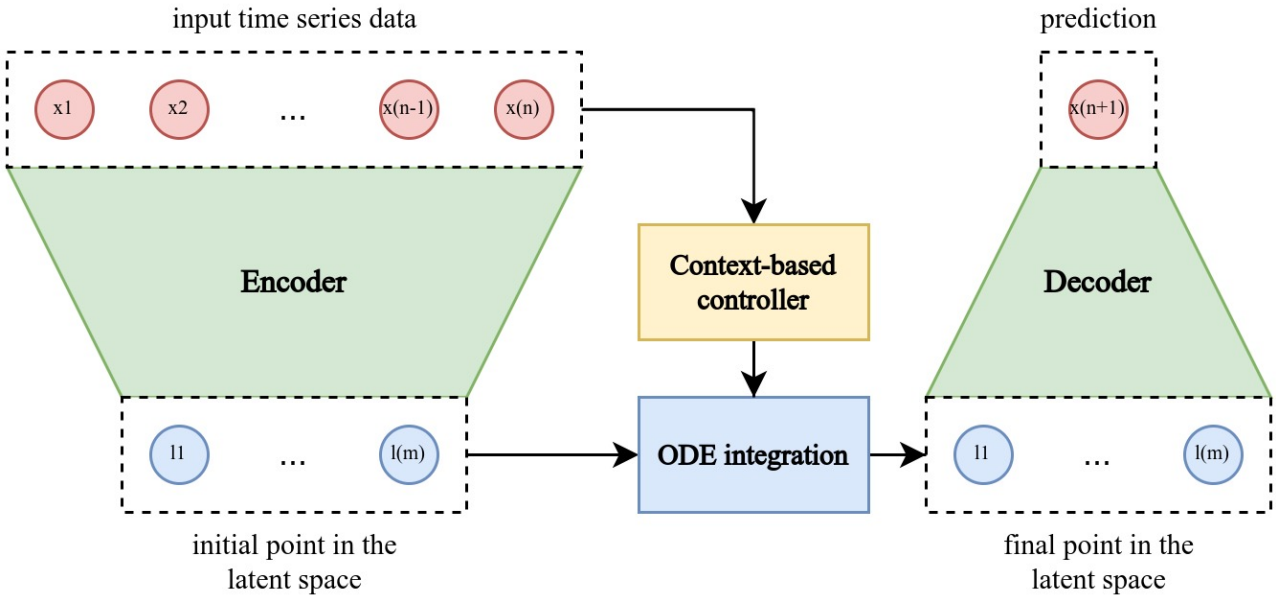


Рис. 2.1: Високорівневе схематичне зображення використання моделі hNODE для передбачення значень одновимірного часового ряду.

2.1 Вибір активаційної функції для нейромережевого моделювання часових рядів

При нейромережевому моделюванні часто використовуються три функції активації: сигмоїда (σ), гіперболічний тангенс (tanh) та зрізаний лінійний вузол (Rectified Linear Unit, ReLU). На широке використання цих функцій є багато причин, однією з яких є числова стійкість, яку вони забезпечують в нейромережевій моделі. Проте існують й інші функції, які забезпечують не гірший рівень стійкості та можуть бути використані для покращення точності моделей.

Розглянемо сімейство степеневих функцій, які можуть використовуватися як активаційні.

Означення 2.2. *[54] Сімейство дійсних функцій $\mathbb{F} \subset \mathbf{C}(\mathbb{X})$ називається розділяючим точки множини $\mathbb{X} \subset \mathbb{R}^n$ якщо для усіх $\mathbf{x}_1, \mathbf{x}_2 \in \mathbb{X}$ $(\mathbf{x}_1 \neq \mathbf{x}_2)$, існує функція $f \in \mathbb{F}$ така, що $f(\mathbf{x}_1) \neq f(\mathbf{x}_2)$.*

Нехай $f(w) \in \mathbb{F}$ – функція дійсної змінної w, така що $f(0) = 0$ і виконуються одна з умов

$$f(w) : \begin{cases} \text{if } w < 0 \text{ then } f(w) = -\psi(-w) < 0, \\ \text{if } w > 0 \text{ then } f(w) = \phi(w) > 0. \end{cases} \tag{2.11}$$

$$f(w) : \begin{cases} \text{if } w < 0 \text{ then } f(w) = \psi(-w) > 0, \\ \text{if } w > 0 \text{ then } f(w) = \phi(w) > 0. \end{cases} \tag{2.12}$$

Тут $\phi(w), \psi(w)$ – диференційовні функції однієї змінної.

Означення 2.3. *[54] Відображення* (2.11)*, (*(2.12)*) називається непарною (парною) функцією активації.*

Розглянемо наступні системи звичайних диференціальних рівнянь:

$$\begin{cases} \dot{x}_1(t) &= f_1(a_{11}x_1 + \ldots + a_{1n}x_n + b_1) \\ \dot{x}_2(t) &= f_2(a_{21}x_1 + \ldots + a_{2n}x_n + b_2) \\ \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \\ \dot{x}_n(t) &= f_n(a_{n1}x_1 + \ldots + a_{nn}x_n + b_n), \end{cases} \tag{2.13}$$

$$\begin{cases} \dot{x}_1(t) &= a_{11}f_1(x_1) + \ldots + a_{1n}f_n(x_n) \\ \dot{x}_2(t) &= a_{21}f_1(x_1) + \ldots + a_{2n}f_n(x_n) \\ \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \\ \dot{x}_n(t) &= a_{n1}f_1(x_1) + \ldots + a_{nn}f_n(x_n), \end{cases} \tag{2.14}$$

де a_{ij}, b_i відомі дійсні константи; $i, j = 1 \ldots n$.

Зауважимо, що у випадку, коли $\det A \neq 0$, систему (2.13) можна перетворити на систему (2.14) і навпаки за допомогою заміни змінних $\mathbf{x} \to A\mathbf{x} + \mathbf{b}$, $A \in \mathbb{R}^{n \times n}$, $\mathbf{b} \in \mathbb{R}^n$. Через це надалі в усіх прикладах будемо використовувати систему (2.14).

Введемо позначення:

$$\mathbf{f}(\mathbf{x}) = (f_1(x_1), \ldots, f_n(x_n))^T. \tag{2.15}$$

Беручи до уваги (2.15), введемо функцію, що містить невизначені інтеграли:

$$V(x_1, \ldots, x_n) = \int f_1(s_1)\, ds_1 + \ldots + \int f_n(s_n)\, ds_n. \tag{2.16}$$

Тоді інтеграл самої функції (2.15) обчислюються за правилами:

1. Якщо $f_i(s_i)$ парне, то

$$\int f_i(s_i)\, ds_i = \begin{cases} \int\limits_0^{x_i} \phi_i(s_i)\, ds_i, \text{ if } x_i \geq 0 \\ \int\limits_{x_i}^0 \psi_i(s_i)\, ds_i, \text{ if } x_i < 0 \end{cases}.$$

2. Якщо $f_i(s_i)$ непарне, то

$$\int f_i(s_i)\, ds_i = \begin{cases} \int\limits_0^{x_i} \phi_i(s_i)\, ds_i, \text{ if } x_i \geq 0 \\ -\int\limits_{x_i}^0 \psi_i(s_i)\, ds_i, \text{ if } x_i < 0 \end{cases}.$$

Теорема 2.1. *[55] Нехай у системі* (2.13) *матриця* $A = \{a_{ij}\}$ *є антисиметричною та невиродженою:* $A^T + A = 0$ *і* $\det(A) \neq 0$. *Також нехай усі функції у векторі* $\mathbf{f}(\mathbf{x})$ *– непарні активаційні функції. Тоді розв'язки* $\mathbf{x}(t, \mathbf{x}_0)$ *системи* (2.13) *обмежені та мають періодичну або хаотичну поведінку.*

Доведення. З [54] відомо, що інтегралом непарної активаційної функції $f_i(x_i), i = 1, \ldots, n$ є парна активаційна функція, а отже $V(x_1, \ldots, x_n) \geq 0$.

Обчислимо повну похідну відносно t функції $V(x_1, \ldots, x_n)$:

$$\begin{aligned} \dot{V}_t(x_1, \ldots, x_n) &= 0.5\left[\dot{x}_1(t)\frac{\partial V(x_1, \ldots, x_n)}{\partial x_1} + \cdots + \dot{x}_n(t)\frac{\partial V(x_1, \ldots, x_n)}{\partial x_n}\right] \\ &\quad +0.5\left[\frac{\partial V(x_1, \ldots, x_n)}{\partial x_1}\dot{x}_1(t) + \cdots + \frac{\partial V(x_1, \ldots, x_n)}{\partial x_n}\dot{x}_1(t)\right] \\ &= 0.5\mathbf{f}^T(\mathbf{x})(A^T + A)\mathbf{f}(\mathbf{x}) \\ &= 0. \end{aligned} \tag{2.17}$$

Тепер, в силу нерівностей (2.11), (2.12), а також того факту, що $V(x_1, \ldots, x_n)$

≥ 0, отримуємо:

$$\lim_{x_i\to\infty}\int\limits_{-x_i}^{x_i} f_i(s_i)\,ds_i = \lim_{x_i\to\infty}\int\limits_{-x_i}^{0} f_i(s_i)ds_i + \lim_{x_i\to\infty}\int\limits_{0}^{x_i} f_i(s_i)\,ds_i \geq 0; i = 1,\ldots,n,$$

що означає $\lim_{\|\mathbf{x}\|\to\infty} V(x_1,\ldots,x_n) \geq 0$. До того ж, з (2.17) випливає, що для достатньо великих значень $\|\mathbf{x}_0\|$ справедлива рівність $V(x_1(t),\ldots,\ x_n(t)) = const = V(x_{10},\ldots,x_{n0}) = V(\mathbf{x}_0) > 0$. Отже, множина

$$\mathbb{S} = \{V(x_1(t),\ldots,x_n(t)) - V(x_{10},\ldots,x_{n0}) = 0\}$$

є компактною, а усі траєкторії $\mathbf{x}(t,\mathbf{x}_0)$ системи (2.13) (або (2.14)) є обмеженими та мають або періодичну, або хаотичну (якщо $n \geq 3$) поведінку. □

Теорема 2.2. *[55] Нехай у системі* (2.13) *матриця* $A+A^T$ *є негативно визначеною. Також нехай усі функції* $\mathbf{f}(\mathbf{x})$ *є непарними активаційними функціями. Тоді розв'язок* $\mathbf{x}(t,\mathbf{x}_0)$ *системи* (2.13) *є стійким.*

Доведення. Очевидно, що $V(0,\ldots,0) = 0$. Тоді, згідно умов з Теореми 2.2, рівність (2.17) має бути замінено нерівністю $\dot{V}_t(x_1,\ldots,x_n) \leq 0$. Тепер залишається лише застосувати теорему Ляпунова про стійкість розв'язків системи звичайних диференціальних рівнянь до системи (2.14). □

Введемо наступні степеневі функції [54]:

$$g(u) = \begin{cases} -(-u)^\beta & \text{if}(u < 0 \text{ and } \beta > 0);\ 0 \text{ if}(u < 0 \text{ and } \beta = 0) \\ u^\alpha & \text{if}(u \geq 0 \text{ and } \alpha > 0);\ 0 \text{ if}(u \geq 0 \text{ and } \alpha = 0) \end{cases} \tag{2.18}$$

або

$$g(u) = \begin{cases} (-u)^\beta & \text{if}(u < 0 \text{ and } \beta > 0);\ 0 \text{ if}(u < 0 \text{ and } \beta = 0) \\ u^\alpha & \text{if}(u \geq 0 \text{ and } \alpha > 0);\ 0 \text{ if}(u \geq 0 \text{ and } \alpha = 0). \end{cases} \tag{2.19}$$

Формула (2.18) є непарною, а (2.19) – парною активаційною функцією. Обидві ці функції мають наступні недоліки з точки зору нейромережевого моделювання:

1. Вони недиференційовні, якщо $0 < \alpha \leq 1$ або $0 < \beta \leq 1$.

2. Вони не приймають до уваги зсув аргументу.

Щоб подолати ці недоліки, введемо наступну функцію:

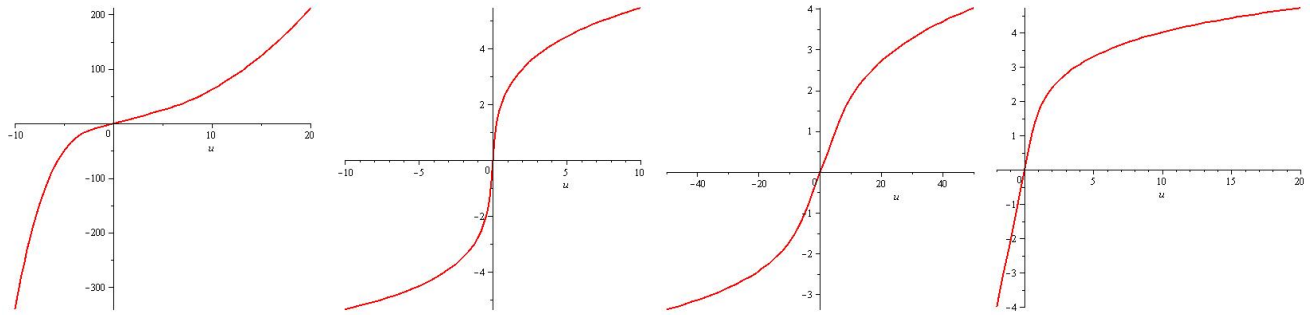
$$w(u,\alpha,\beta,b,c) = \begin{cases} -\dfrac{\beta-1}{\beta}c^{\frac{\beta}{\beta-1}} - \dfrac{1}{\beta}\Big(-\Big(u+\dfrac{b}{c}\Big)\Big)^{\beta} & \text{, if } u+\dfrac{b}{c} < -c^{\frac{1}{\beta-1}} \\ c\cdot\Big(u+\dfrac{b}{c}\Big) & \text{, if } u+\dfrac{b}{c} \le c^{\frac{1}{\alpha-1}} \\ \dfrac{\alpha-1}{\alpha}c^{\frac{\alpha}{\alpha-1}} + \dfrac{1}{\alpha}\Big(u+\dfrac{b}{c}\Big)^{\alpha} & \text{, otherwise.} \end{cases} \tag{2.20}$$

Тут $\alpha > 0$, $\beta > 0$, $\alpha \neq 1$, і $\beta \neq 1$ степені, $c > 0$ тангенс кута нахилу прямої лінії $w = cu + b$, а b – зсув аргументу.

Поклавши в (2.20) $b = 0$, отримаємо:

$$w(u,\alpha,\beta,b,c) = \begin{cases} -\dfrac{\beta-1}{\beta}c^{\frac{\beta}{\beta-1}} - \dfrac{(-u)^{\beta}}{\beta} & \text{, if } u < -c^{\frac{1}{\beta-1}} \\ c\cdot u & \text{, if } u \le c^{\frac{1}{\alpha-1}} \\ \dfrac{\alpha-1}{\alpha}c^{\frac{\alpha}{\alpha-1}} + \dfrac{u^{\alpha}}{\alpha} & \text{, otherwise.} \end{cases} \tag{2.21}$$

Формула (2.21) також може бути отримана з (2.20) шляхом введення нової змінної $z = u + b/c$, яка в (2.21) знову позначена як $u := z$. Графічний вигляд функції (2.21) наведено на Рис. 2.2.



(а) (б) (в) (г)

Рис. 2.2: Диференційовні степеневі функції активації (2.21) для різних значень параметрів α, β, і c: (a) $c = 5, \alpha = 2, \beta = 3$; (b) $c = 2, \alpha = 0.01, \beta = 2$; (c) $c = 7, \alpha = 0.3, \beta = 0.1$; (d) $c = 0.2, \alpha = 0.1, \beta = 0.01$.

Під час оптимізації моделі, що використовує функцію (2.21), важливо знати її похідну. У випадку, коли $\alpha > 0$, $\beta > 0$, $\alpha \neq 1, \beta \neq 1$, і $c \geq 0$, похідна від

$w(u, \alpha, \beta, c)$ має наступний вигляд:

$$\dot{w}_u(u, \alpha, \beta, c) = \begin{cases} (-u)^{\beta-1} & \text{, if } u < -c^{\frac{1}{\beta-1}} \\ c & \text{, if } u \leq c^{\frac{1}{\alpha-1}} \\ u^{\alpha-1} & \text{, otherwise.} \end{cases} \tag{2.22}$$

Якщо $\lim \beta \to 1$, то

$$w_u(u, \alpha, \beta, c) \to \begin{cases} c \cdot u & \text{, if } u \leq c^{\frac{1}{\alpha-1}} \\ \dfrac{\alpha-1}{\alpha} c^{\frac{\alpha}{\alpha-1}} + \dfrac{u^\alpha}{\alpha} & \text{, otherwise.} \end{cases}$$

$$\dot{w}_u(u, \alpha, \beta, c) \to \begin{cases} c & \text{, if } u \leq c^{\frac{1}{\alpha-1}} \\ u^{\alpha-1} & \text{, otherwise.} \end{cases}$$

Якщо $\lim \alpha \to 1$, то

$$w(u, \alpha, \beta, b, c) \to \begin{cases} -\dfrac{\beta-1}{\beta} c^{\frac{\beta}{\beta-1}} - \dfrac{(-u)^\beta}{\beta} & \text{, if} u < -c^{\frac{1}{\beta-1}} \\ c \cdot u & \text{, otherwise.} \end{cases}$$

$$\dot{w}(u, \alpha, \beta, b, c) \to \begin{cases} (-u)^{\beta-1} & \text{, if } u < -c^{\frac{1}{\beta-1}} \\ c & \text{, otherwise.} \end{cases} \tag{2.23}$$

Якщо $\lim \alpha \to 1$ і $\lim \beta \to 1$, то $w(u, \alpha, \beta, c) \to cu$ і $\dot{w}_u(u, \alpha, \beta, c) \to c$.

Зауважимо, що рівняння (2.21) перетворюється на (2.20), якщо покласти в (2.21) $u := u + b/c$, тобто $w(u + b/c, \alpha, \beta, c) \equiv w(u, \alpha, \beta, b, c)$.

Нарешті, якщо покласти $c = 0$ в рівнянні (2.21) та виконати деякі незначні перетворення, отримаємо функцію (2.18):

$$w(u, \alpha, \beta) = \begin{cases} -\dfrac{(-u)^\beta}{\beta} & \text{, if } u < 0 \\ \dfrac{u^\alpha}{\alpha} & \text{, otherwise.} \end{cases} \tag{2.24}$$

$$\dot{w}(u, \alpha, \beta) = \begin{cases} (-u)^{\beta-1} & \text{, if } u < 0 \\ u^{\alpha-1} & \text{, otherwise,} \end{cases} \tag{2.25}$$

де $\alpha > 1$ і $\beta > 1$.

Також зауважимо, що функції (2.20) і (2.21) диференційовні всюди на ін-

тервалі $(-\infty, \infty)$ та за будь-яких $\alpha > 0$ і $\beta > 0$. Водночас функція (2.24) недиференційовна за $0 < \alpha \leq 1$ або $0 < \beta \leq 1$, в точці $u = 0$.

Таким чином, функції (2.20) і (2.21) є узагальненням непарної степеневої функції активації (2.18). Узагальнення полягає в тому, що, на відміну від (2.18), (2.21) є всюди диференційовною, а отже є оптимальною для використання в градієнтних методах оптимізації.

Як приклад використання (2.21), наведемо узагальнену функцію активації з кубічним коренем, яка є диференційовною на усій осі дійсних чисел. Для цього використаємо функції (2.21) та (2.22) з параметрами $\alpha = \beta = 1/3$ і $c = 1$:

$$w(u, \alpha, \beta, b, c) = \begin{cases} 2 - 3(-u)^{1/3} & \text{, if } u < -1 \\ u & \text{, if } u \leq 1 \\ -2 + 3u^{1/3} & \text{, otherwise.} \end{cases}$$

Похідна функції $w(u)$ неперервна, але не диференційовна, та обмежена ($\dot{w}_u(u) \in (0, 1]$) на усьому проміжку $(-\infty, \infty)$. Графік $w(u)$ та її похідної наведено на Рис. 2.3.

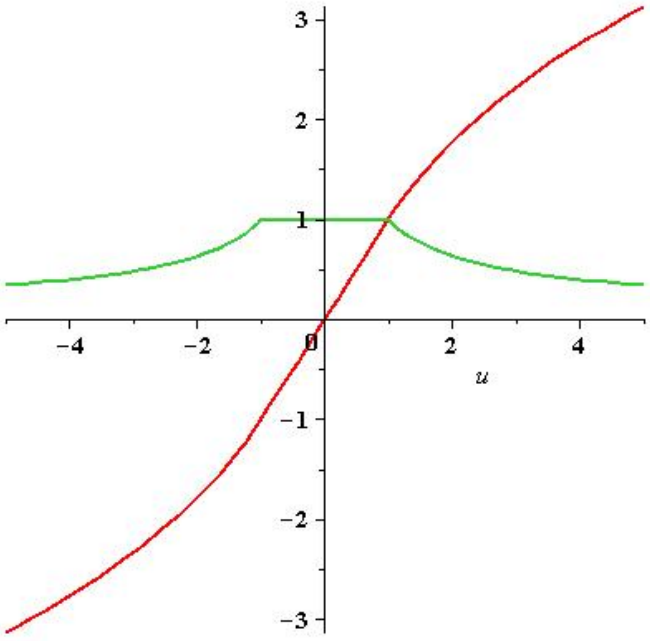


Рис. 2.3: Степенева функція активації та її похідна $w(u)$ (червона), $\dot{w}_u(u)$ (зелена)

2.2 Прогнозування часових рядів без явної параметризації

Як згадувалося раніше, головним недоліком базової моделі NODE є те, що він накладає жорсткі вимоги на розмірність моделі та вимагає, щоб породжені ним траєкторії існували у тому ж просторі, що і вхідні дані. Тобто у випадку моделювання одновимірних часових рядів модель також має бути одновимірною, а оскільки NODE блок є, по суті, автономною системою ЗДР, в одновимірній формі він буде обмежений усього трьома можливими видами поведінки. Кожна з цих поведінок напряму залежить від показника Ляпунова, жоден з яких не дозволяє моделі мати періодичні або квазіперіодичні траєкторії. Одним зі способів для уникнення цієї проблеми є перехід від одновимірного ряду до багатовимірного шляхом побудови його вкладення у більш високовимірному просторі. Для цього необхідно оцінити мінімальну розмірність d простору вкладення та значення часів затримки $\{\tau_1, \tau_2, \ldots, \tau_{d-1}\}$, використовуючи один з відомих алгоритмів для побудови вкладення [35], [56]. Ця операція надасть моделі необхідний контекст для коректної роботи шляхом побудови множини унікальних комбінацій точок $U^* \in \mathbb{R}^d$, $U^* = \left\{u_n, u_{n-\tau_1}, \ldots, u_{n-\tau_{d-1}}\right\}$, які можна використовувати для побудови авторегресійних моделей вигляду $u_{n+1} = f(U^*, \Theta)$.

Проте результат моделі на основі NODE у цьому випадку буде мати розмірність d, а не 1. Цю проблему можна частково вирішити, якщо відкинути виміри, що не відповідають прогнозованим значенням оригінального часового ряду. Проте такий підхід накладе додаткові обмеження на модель та фактично змусить основну систему ЗДР будувати одну і ту саму траєкторію, що відрізняється тільки часом затримки, всіма її координатами. Більшість високовимірних систем ЗДР за нормальних умов так не поводяться. Щоб розв'язати цю проблему більш коректно, можна дозволити моделі працювати у деякому латентному просторі L, у якому не буде подібних обмежень. Точки у цьому латентному просторі будуть отримуватися за допомогою окремої моделі-кодувальника $\mathcal{E}$, яка не тільки застосує затримку в часі, але і проведе додаткові перетворення для декореляції компонентів нового вектора вкладення U^*. Використання кодувальника $\mathcal{E}$ пом'якшує жорстку залежність стану NODE від даних часового ряду, але також необхідно інтерпретувати точки в латентному просторі L для отримання прогнозованих даних u_{n+i}. Для цього використаємо ще одну

модель-декодувальник $\mathcal{D}$, яка перетворить точки траєкторії NODE у скаляри, що існують в тому ж просторі, що і модельований часовий ряд.

Кодувальник і декодувальник прибирають будь-які обмеження з моделі NODE та дозволяють вільно обирати структуру та розмірність системи ЗДР, яка у ній прихована. Для моделювання одновимірних часових рядів, які можуть мати періодичну або квазіперіодичну поведінку, доцільно обирати структуру моделі так, щоб вона могла коректно відображати цю поведінку. Однією з таких структур є антисиметрична рекурентна нейронна мережа (AntisymmetricRNN) [57], яка задається рівнянням:

$$\mathbf{h}_n = \mathbf{h}_{n-1} + \varepsilon\sigma((\mathbf{W_h} - \mathbf{W_h}^T - \gamma\mathbf{I})\mathbf{h}_{n-1} + \mathbf{V}_h x_n + \mathbf{b_h}), \tag{2.26}$$

де $\mathbf{h}$ – прихований стан, $\mathbf{x}$ – вхідні дані, а σ – функція активації.

Головна мета антисиметричної RNN у тому, щоб структурно нав'язати моделі періодичну поведінку траєкторій. Це досягається за допомогою конструювання матриці параметрів так, щоб власні значення її якобіану мали нульові або дещо негативні дійсні частини. Модель (2.26) структурована так, щоб її прихований стан не зростав та не зменшувався швидко при переході від однієї точки вхідних даних до іншої. Модель hNODE не має прихованого стану, а процеси, для моделювання яких вона може застосовуватись, можуть мати дрейфувальну поведінку, що може означати наявність власних значень якобіана з позитивною дійсною частиною. Враховуючи ці зауваження, визначимо приховану систему ЗДР у рівнянні (2.10) як

$$G(l, \mathbf{W}, \mathbf{D}, \mathbf{b}) = l_0 + \int_0^T \sigma((\mathbf{W} - \mathbf{W}^T + \mathbf{D})l + \mathbf{b})\, dt. \tag{2.27}$$

Тут $\mathbf{W}$ – це матриця з необмеженими значеннями, яка перетворюється на антисиметричну шляхом віднімання її транспонованої версії від самої себе; $\mathbf{D}$ – діагональна матриця; $\mathbf{b}$ – вектор зсуву; l – точка у латентному просторі. Усі фактичні значення параметрів отримуються під час обчислення гіпермоделлю $\mathcal{H}(X, \Theta)$ (див. (2.9)).

Враховуючи можливість нестабільної поведінки або якісних змін, що можуть виникнути у модельованому часовому ряді, модель (2.27) повинна мати змогу адаптуватися до таких змін. Для цього можемо використати функцію $\mathcal{H}$, яка б генерувала новий вектор параметрів залежно від поточного контексту

кожного разу, коли обчислюється функція (2.27). Це дозволяє hNODE змінювати поведінку залежно від якісних характеристик поточного сегмента часового ряду, що подається на вхід моделі. Активаційна функція σ у наведених нижче прикладах обрана у вигляді (2.21) з параметрами $\alpha = 0.5$, $\beta = 0.3$, і $c = 5$. Графік активаційної функції та її похідної наведено на Рис. 2.4.

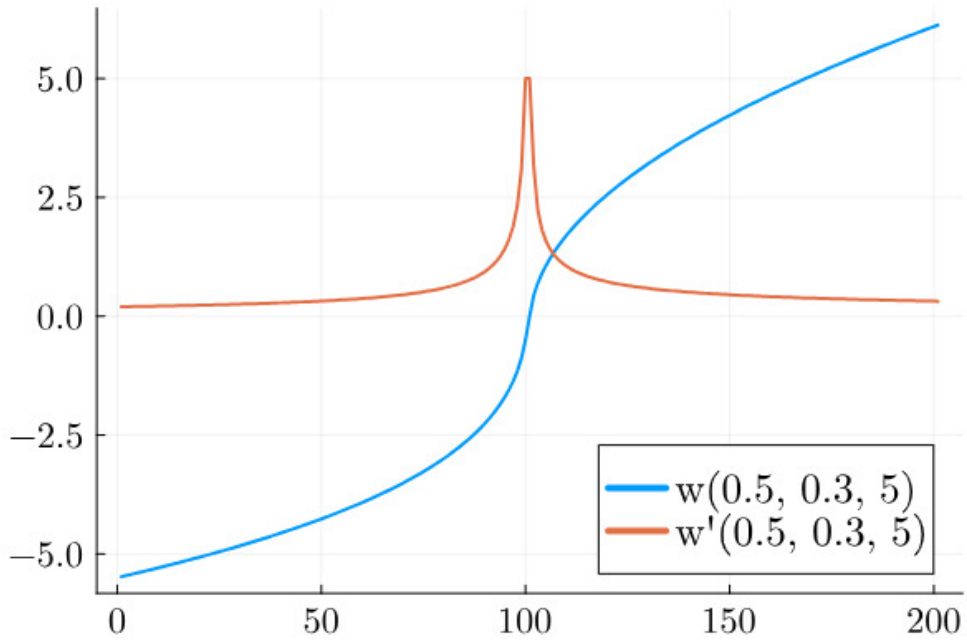
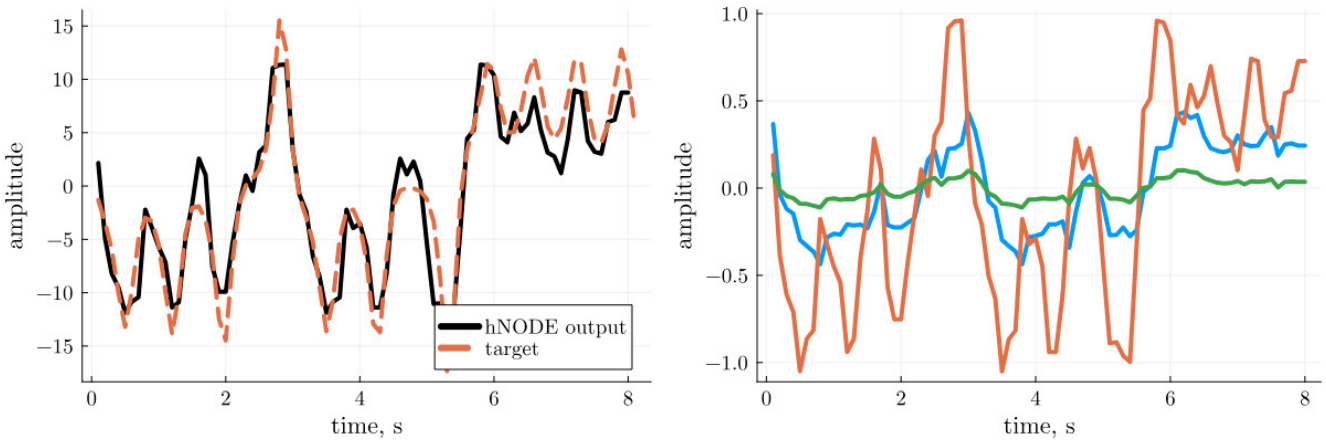


Рис. 2.4: Неперервно диференційовна функція активації та її недиференційовна перша похідна.

Фінальна архітектура hNODE з усіма компонентами зібраними в одній моделі наведено на Рис. 2.1.

Модель з архітектурою, описаною вище, використаємо для передбачення значень першої координати хаотичної траєкторії системи Лоренца (3.18). Значення координати x були вкладені у простір розмірністю $d = 3$ та затримкою $\tau = 1$. Цей набір даних потім був розбитий на сегменти довжиною $d \cdot \tau + M$ $(M \geq 1)$ та сформований як $\{X_m, Y_m\}$, $m \in [(d-1) \cdot \tau, n - M]$, $X_m = \{x_m, x_{m-\tau}, \ldots, x_{m-(d-1)\cdot\tau}\}$, $Y_m = \{x_{m+1}, x_{m+2}, \ldots, x_{m+M}\}$. Тобто перші $d \cdot \tau$ точок із вкладеної траєкторії були обрані у якості вхідного контексту, а решта M точок – очікуваною траєкторією моделі.

Обчислення результату моделі виконувалося рекурсивно – вхідні дані були використані для отримання прогнозу наступної точки у часовому ряді, після чого подовжений прогнозом часовий ряд було використано для прогнозування ще однієї точки, і так далі. Цей процес використовувався доки не було отримано прогнозовану траєкторію бажаної довжини.



(а) (б)

Рис. 2.5: hNODE генерує траєкторію, подібну до оригінальної системи (a) з латентних траєкторій вкладеної NODE (b).

Таким чином, модель hNODE здатна самостійно визначити параметри для вкладеної NODE виключно з даних часового ряду та без додаткових сторонніх входів.

2.3 Прогнозування часових рядів з явною параметризацією шляхом включення параметрів в контекст

Оскільки дані часових рядів повністю відокремлені від латентного простору, в якому еволюціонують траєкторії вкладеної моделі NODE, ці дані можна розширити, включивши до них контекст, який має допомогти моделі досягти бажаної поведінки. Наприклад, модель hNODE можна натренувати таким чином, щоб вона генерувала синусоїду певної частоти, значення якої передавалося б до функції $\mathcal{H}$ разом з поточним контекстом.

Як і у попередньому підрозділі, структура вкладеної NODE була задана моделлю (2.27). Часовий ряд був згенерований синусоїдами різних частот, дискретизованими з частотою 8 кГц та вкладеному у простір з розмірністю $d = 2$ і часом затримки $\tau = 1$. Результат вкладення був підготовлений для тренування моделі так само як і тренувальні дані для прогнозування атрактора системи Лоренца, але кожен вектор входу X_m був доповнений значенням бажаної частоти F_0, тобто вхідні дані мали вигляд $X_m = \{x_m, x_{m-\tau}, F_0\}$. Ще однією відмінні-

стю цього прикладу від попереднього є те, що результат моделі отримувався шляхом обчислення повної траєкторії у латентному просторі та декодуванням усіх точок одразу замість рекурсивного застосування моделі до прогнозованого часового ряду. Це демонструє другий спосіб використання моделі hNODE, який так само дає задовільні результати. Результати роботи моделі для різних параметрів F_0 наведено на Рис. 2.6.

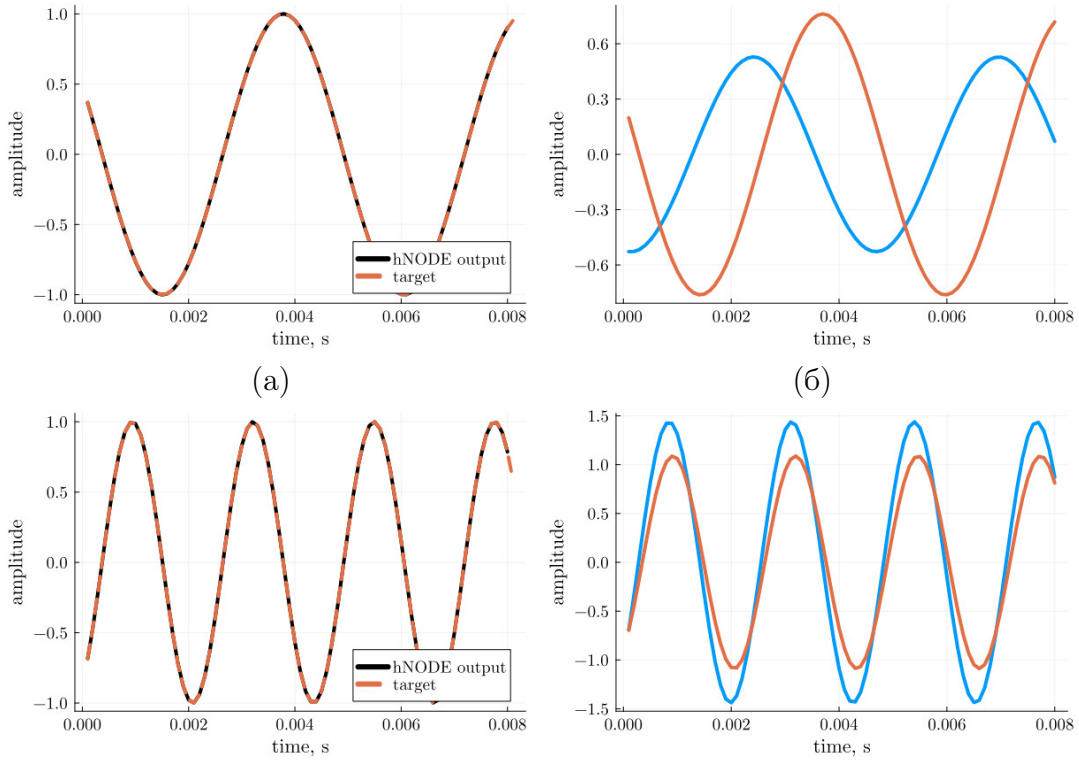


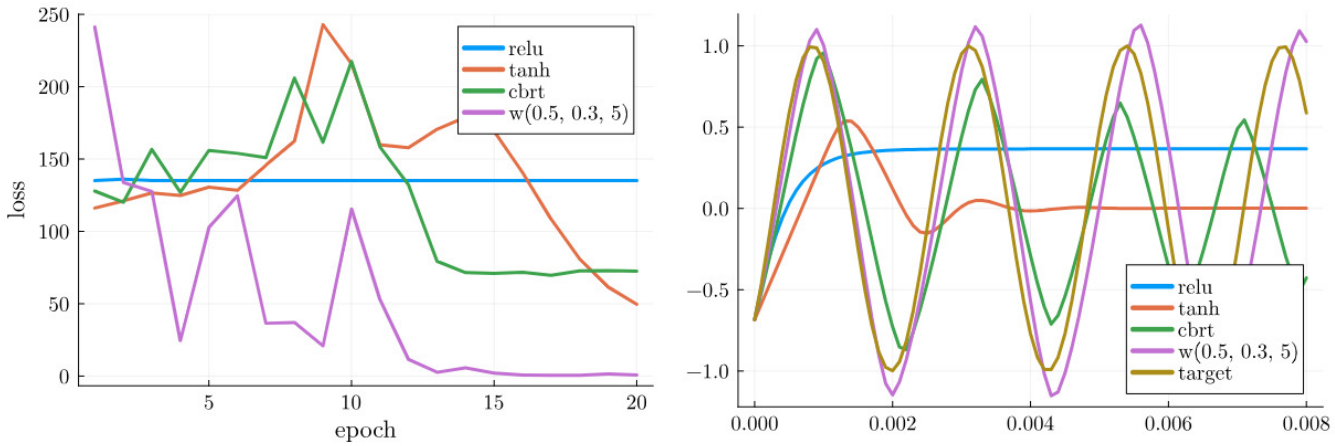
Рис. 2.6: Модель hNODE генерує синусоїди для явно заданого параметра частоти. Вивід моделі наведено на (a, c), а латентні траєкторії вкладеної NODE – на (b, d).

2.4 Порівняння впливу різних активаційних функцій на якість прогнозування

Як зазначалося у попередніх підрозділах, вибір функції активації є дуже важливим при побудові нейромережевої моделі. Неправильно вибрана активаційна функція може сповільнити темпи навчання або взагалі зробити модель непридатною для розв'язання даної задачі. Щоб продемонструвати це, було обрано простий приклад схожий на той, що наведено у попередньому підрозділі

– звичайна синусоїда, яку модель hNODE має передбачити без включення додаткових параметрів до контексту. До порівняння були включені розповсюджені функції активації – ReLU та гіперболічний тангенс (tanh) – а також кубічний корінь ($cbrt(x) = \sqrt[3]{x}$) та функція (2.21) з параметрами $\alpha = 0.5$, $\beta = 0.3$, та $c = 5$.

Користуючись тим фактом, що модель hNODE складається з декількох окремих моделей, кодувальник $\mathcal{E}$ та декодувальник $\mathcal{D}$ було натреновано окремо. Потім модель-кодувальник було використано для побудови траєкторій в латентному просторі, на яких в підсумку і тренувалась вкладена NODE модель за принципом, описаних у попередніх підрозділах. Кожна NODE у цьому порівнянні була натренована на двох наборах даних, які відрізнялися довжиною очікуваних траєкторій. Перша група містила сегменти латентної траєкторії довжиною 5, а друга – 50. Темп навчання ε для першої групи був обраний як 0.01, а для другої – 0.001. Кожна модель була послідовно натренована на цих групах впродовж десяти епох. Результати тренування всіх моделей наведені на Рис. 2.7.



(а) (б)

Рис. 2.7: Значення функції втрат протягом епох тренування для різних функцій активації (a) та траєкторії прогнозів, отримані з використанням цих функцій (b).

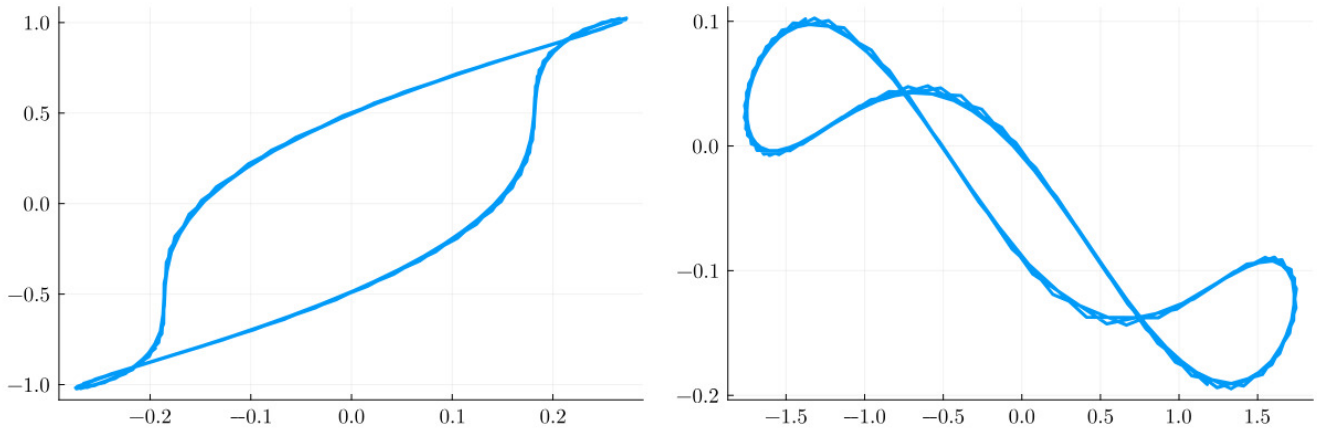
Як видно з рисунків, зі степеневою функцією активації (2.21) було досягнуто найкращих результатів – модель, натренована з її використанням, показала найкращу точність та зійшлася до оптимальних значень параметрів найшвидше. Другою за якістю передбачень виявилася функція гіперболічного тангенса, яка забезпечила швидке сходження наприкінці тренування. Функція кубічного кореня зійшлася майже одночасно зі степеневою, але зі значно гіршими значен-

нями втрат на тестових даних. Нарешті, ReLU виявилася найгіршою та непридатною для розв'язання цієї задачі.

2.5 Можливості для виникнення некоректної поведінки під час тренування моделі hNODE

Хоча модель hNODE є доволі гнучкою та може бути застосована для розв'язання багатьох різних задач, є деякі застереження щодо її використання. Ці застереження здебільшого пов'язані з природою вкладеної в неї NODE моделі.

По-перше, траєкторії в латентному просторі, породжені з даних часового ряду моделлю-кодувальником, не мають перетинатися, оскільки це порушило б умови про існування та єдиність розв'язку звичайних диференціальних рівнянь, які модель намагається наближено представити. Приклад неправильно сформованих траєкторій у латентному просторі наведено на Рис. 2.8.



(а) (б)

Рис. 2.8: Перетин у двовимірному просторі траєкторій, отриманих шляхом вкладання сегментів синуса в латентний простір моделлю-кодувальником.

По-друге, крок інтегрування, що використовується при чисельному розрахунку траєкторій, має бути обраний відповідно до даної задачі. На перший погляд здається небезпідставним обрати за крок інтегрування значення часового кроку Δt самого часового ряду. Наприклад, можна використати обернене значення частоти дискретизації синусоїди з попередніх прикладів, але навіть для доволі невисокої частоти дискретизації такий крок виявляється замалим для забезпечення нормального режиму тренування моделі – для частоти дискретизації 8 кГц він становить всього 0.000125. Ситуацію з неправильно обраним

кроком інтегрування непросто виявити, і вона може суттєво погіршити темпи навчання моделі. Приклад використання різних кроків інтегрування та порівняння якості роботи моделі на прикладі з попереднього підрозділу зі степеневою функцією активації наведено на Рис. 2.9.

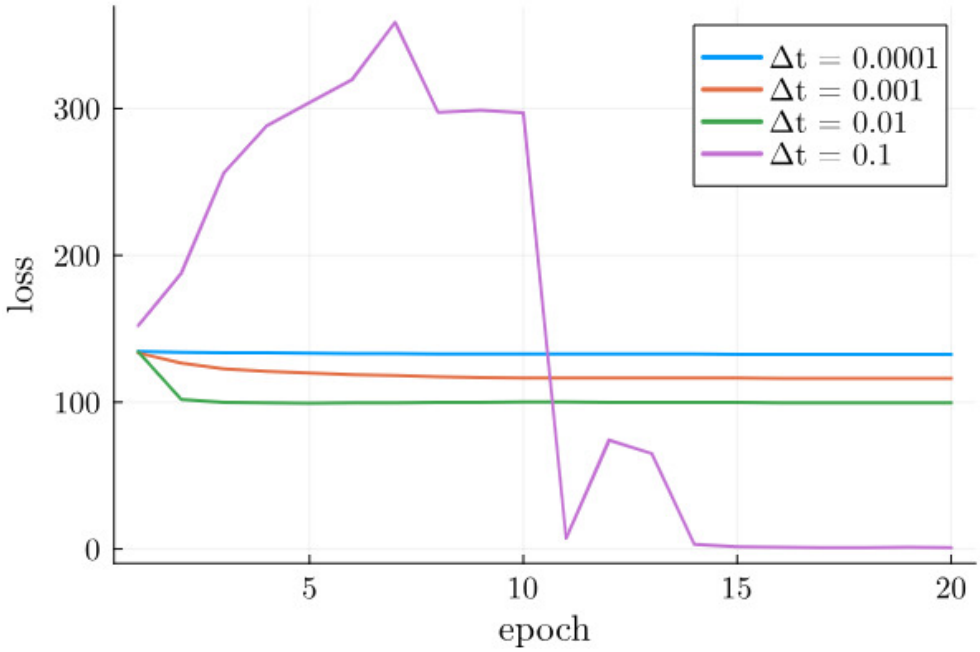


Рис. 2.9: Вплив кроку інтегрування Δt на швидкість та якість навчання моделі.

2.6 Обґрунтування вибору структури моделі та застосованих в ній активаційних функцій

При побудові нейромережевих архітектур, часто використовують випробувані часом активаційні функції, такі як гіперболічний тангенс, сигмоїда, або ReLU. Однак ці функції можуть виявитися неоптимальними для даної задачі. Гіперболічний тангенс та сигмоїда диференційовні, але обмежені, що може привести до проблеми зникання градієнта. ReLU, своєю чергою, необмежена, але недиференційовна в точці 0, що може призвести до неоптимального тренування моделі в деяких оптимізаторах. Замість гіперболічного тангенса або сигмоїди можна використати схожі за формою функції, наприклад, кубічних корінь, але він, як і ReLU, недиференційовний в точці 0. До необмежених та всюди диференційовних функцій належить тотожне відображення, яким часто нехтують при побудові моделей. Саме по собі воно не надає моделі нелінійних властивостей, але може бути використане для побудови поясненних моделей, а також

скоротити час їх тренування завдяки відсутності витрат на обчислення такої функції. Нелінійних властивостей можна досягти, обравши спеціальні правила обчислення моделі. У цьому розділі розглядається спеціально розроблена функція активації, що є необмеженою та диференційовною, а також якість її роботи у порівнянні з тотожним відображенням, а також комбінація цих функцій в одній моделі у контексті розв'язку задачі апроксимації одновимірних часових рядів із застосуванням архітектури hNODE.

У сучасних дослідженнях динамічних процесів надається перевага екстенсивній радше ніж інтенсивній розробці нейромережевих моделей для розв'язку різноманітних проблем, що стосуються виявлення прихованих характеристик таких процесів. Такий підхід є цілком виправданим, оскільки він покладається на той факт, що глибокі нейромережі здатні досягати експоненційної апроксимації для достатньо гладких функцій [58]—[61]. Проте більшість проривів в галузі нейромережевого моделювання були здійснені саме моделями, архітектури яких були розроблені дослідниками вручну та містили бажані характеристики ще на етапі конструювання. Прикладами таких моделей є згорткові мережі [62], [63], залишкові мережі [64], [65], рекурентні мережі [66], [67], мережі з коротко- та довгостроковою пам'яттю [68], [69], трансформери [70], [71], нейронні ЗДР [43], [72], та моделі з простором вибіркових станів [73].

Усі зазначені вище моделі беруть до уваги як просторові, так і часові характеристики модельованих даних, а також є достатньо гнучкими для їх застосування у різних прикладних задачах. Однак для подальшого покращення точності моделі у конкретній задачі, можна ввести в неї додаткові обмеження, що забезпечать збереження бажаних інваріантів. Такий клас моделей називають фізично поінформованими (physics-informed NN, PINN) [74]—[76]. Проте інваріанти часто невідомі. У цьому випадку, бажаним є створення пояснюваних моделей, що дозволять, шляхом дослідження їх властивостей, знаходити невідомі інваріанти. Однією з таких моделей є hNODE.

2.6.1 Обмеженість розв'язків системи нейронних ЗДР

Нехай $\mathbf{u} \in \mathbb{R}^n$ – дійсний вектор розмірністю n, а $\sigma(cu)$ – дійсна функція однієї змінної, яка є або гіперболічним тангенсом, або степеневою функцією активації $w(u, \alpha, \beta, c)$.

Введемо векторну форму функції $\sigma(cu)$, замінивши скаляр cu векторною

змінною $W\mathbf{u}$:

$$\sigma(cu) \to \sigma(W\mathbf{u}) = (\sigma(\mathbf{w}_1, \mathbf{u}), ..., \sigma(\mathbf{w}_n, \mathbf{u}))^T \in \mathbb{R}^n,$$

де $W = (\mathbf{w}_1^T, ..., \mathbf{w}_n^T)^T$. Таке представлення будемо використовувати, якщо $\sigma(cu) = w(u, \alpha, \beta, c)$. У разі, коли $\sigma(cu) = tanh(cu)$, скалярний параметр c перетворюється на матрицю $c \to W \in \mathbb{R}^{n\times n}$.

Тепер розглянемо систему звичайних диференціальних рівнянь, що є узагальненням моделі (2.27):

$$\dot{\mathbf{x}}(t) = A\mathbf{x}(t) + \mathbf{c} + \sigma(W\mathbf{x}(t) + \mathbf{b}), \tag{2.28}$$

де $A, W \in \mathbb{R}^{n\times n}$ та $\mathbf{c}, \mathbf{b} \in \mathbb{R}^n$.

Теорема 2.3. *[77] (Про стискаюче відображення). Нехай $\mathbb{S} \subset \mathbb{R}^n$ – непорожня компактна множина і $\mathbf{T}$ – неперервне відображення з $\mathbb{S}$ в $\mathbb{S}$. Тоді якщо*

$$\|\mathbf{T}(\mathbf{x}) - \mathbf{T}(\mathbf{y})\| \le \rho\|\mathbf{x} - \mathbf{y}\|, \forall \mathbf{x}, \mathbf{y} \in \mathbb{S}, 0 < \rho < 1,$$

то існує унікальний вектор $\mathbf{u}^* \in \mathbb{S}$ такий, що $\mathbf{u}^* = \mathbf{T}(\mathbf{u}^*)$.

Лема 2.1. *Нехай $\sigma(cu) = \tanh(cu)$. Якщо матриці $\boldsymbol{A}$ і $\boldsymbol{W}$ невироджені та $0 < c \le 1$, то система (2.28) має хоча б одну точку рівноваги.*

Доведення. Нехай $c = 1$. Покажемо, що система рівнянь

$$A\mathbf{x} + \mathbf{c} + \sigma(W\mathbf{x} + \mathbf{b}) = \mathbf{0} \tag{2.29}$$

має дійсний розв'язок.

Введемо позначення $\mathbf{u} = -A\mathbf{x} - \mathbf{c} \Rightarrow \mathbf{x} = -A^{-1}(\mathbf{u} + \mathbf{c})$ та перепишемо систему (2.28) як:

$$\mathbf{u} = \sigma(-WA^{-1}\mathbf{u} - WA^{-1}\mathbf{c} + \mathbf{b}) = \sigma(M\mathbf{u} + \mathbf{g}), \tag{2.30}$$

де $M = -WA^{-1} = \{m_{ij}\}$ та $\mathbf{g} = -WA^{-1}\mathbf{c} + \mathbf{b} = \{g_i\}$, $i, j = 1, \ldots, n$.

Перепишемо рівняння (2.30) у наступній формі:

$$\mathbf{u} = \mathbf{u} - \frac{1}{\mu}\Big(\mathbf{u} - \sigma(M\mathbf{u} + \mathbf{g})\Big) = \mathbf{T}(\mathbf{u}), \mu > 1, \tag{2.31}$$

та введемо векторну норму $\|\mathbf{u}\| = \max_{1\le i\le n} |u_i|$.

Тепер скористаємося результатами аналізу з [77].

Відрізок $L(\mathbf{x},\mathbf{y})$, що сполучає вектори $\mathbf{x},\mathbf{y}\in\mathbb{R}^n$, задамо як

$$L(\mathbf{x},\mathbf{y}) = \{\mathbf{z} \mid \mathbf{z} = \theta\mathbf{x} + (1-\theta)\mathbf{y}, 0<\theta<1\}.$$

Нехай дійсна функція $f:\mathbb{R}^n\to\mathbb{R}$ неперервно диференційовна в будь-якій точці множини $\mathbb{S}$ та $\mathbf{x},\mathbf{y}$ належать $\mathbb{S}$ так, що $L(\mathbf{x},\mathbf{y})\subset\mathbb{S}$. Тоді існує точка $\mathbf{z}\in L(\mathbf{x},\mathbf{y})$, що

$$f(\mathbf{x}) - f(\mathbf{y}) = \frac{\partial f}{\partial \mathbf{x}}\Big|_{\mathbf{x}=\mathbf{z}} (\mathbf{x}-\mathbf{y}).$$

Оскільки $||\mathbf{x}||$ – неперервна функція, то застосувавши її до попереднього рівняння і використавши Теорему 2.3 для рівняння (2.31), отримаємо:

$$\begin{aligned}
&\Big\|\Big(1-\frac{1}{\mu}\Big)\mathbf{u} + \frac{1}{\mu}\sigma(M\mathbf{u}+\mathbf{g}) - \Big(1-\frac{1}{\mu}\Big)\mathbf{w} - \frac{1}{\mu}\sigma(M\mathbf{w}+\mathbf{g})\Big\| \\
&\quad\le \Big\|\Big(1-\frac{1}{\mu}\Big)(\mathbf{u}-\mathbf{w})\Big\| + \frac{1}{\mu}\|\dot\sigma(\mathbf{z})\|\cdot\|M\mathbf{u}-M\mathbf{w}\| \\
&\quad\le \Big(1-\frac{1}{\mu}+\frac{1}{\mu}\|\dot\sigma(\mathbf{z})\|\cdot\|M\|\Big)\cdot\|\mathbf{u}-\mathbf{w}\| = \rho\cdot\|\mathbf{u}-\mathbf{w}\|,
\end{aligned}$$

де

$$\rho = 1 - \frac{1}{\mu} + \frac{1}{\mu}\|\dot\sigma(\mathbf{z})\|\cdot\|M\|.$$

Звідси маємо, що якщо $\|\dot\sigma(\mathbf{z})\|\cdot\|M\| = \|\dot\sigma(\mathbf{z})\|\cdot\|WA^{-1}\| < 1$, то $\rho<1$.

Функція $\sigma(x)$ задовольняє нерівності $0\le\dot\sigma(x)\le 1$, $\forall x\in\mathbb{R}$. І $\lim\limits_{|x|\to\infty}\dot\sigma(x) = 0$ та якщо $|x|\ne 0$, то $\dot\sigma(x)\ne 0$.

Отже, а) якщо $\|M\|<1$, то $\|\dot\sigma(\mathbf{z})\|<1$ і $\rho<1$; б) якщо $\|M\|>1$ і $\|\dot\sigma(\mathbf{z})\| < \|M\|^{-1}$, то також $\rho<1$

Розглянемо шар $\mathbb{S} := \{\mathbf{u}\in\mathbb{R}^n \mid \|\mathbf{u}-\mathbf{p}\|^2\le 1\}$, де $\mathbf{p} = -M^{-1}\mathbf{g} = -AW^{-1}\mathbf{g}$. Оберемо вектор $\mathbf{v}\in\mathbb{S}$ такий, що $\mathbf{v}\approx -M^{-1}\mathbf{g}$, тоді:

$$\mathbf{T}(\mathbf{v}) = \mathbf{v} - \frac{1}{\mu}\Big(\mathbf{v} - \sigma(M\mathbf{v}+\mathbf{g})\Big)$$

та

$$\|\mathbf{T}(\mathbf{v})\| \le \Big\|\mathbf{v} - \frac{1}{\mu}\Big(\mathbf{v} - \sigma(M\mathbf{v}+\mathbf{g})\Big)\Big\|$$
$$\le \Big(1-\frac{1}{\mu}\Big)\|\mathbf{v}\| + \frac{1}{\mu}\Big\|\sigma(M\mathbf{v}+\mathbf{g})\Big\| \le \Big(1-\frac{1}{\mu}\Big)\|\mathbf{v}\| + \frac{\delta}{\mu} \le \|\mathbf{v}\|,$$

де $\delta = \lim\limits_{\mathbf{v}\to\mathbf{p}} \|\sigma(M\mathbf{v}+\mathbf{g})\| = 0$ та значення $\mu > 1$ є достатньо великим.

а) $\|M\| < 1$. У цьому випадку $\forall \mathbf{u} \in \mathbb{S}$ і $\|\dot\sigma(\mathbf{z})\| \cdot \|M\| < 1$, а отже $\mathbf{T}(\mathbf{u}) \in \mathbb{S}$.

б) $\|M\| > 1$. Цей випадок довести складніше. Очевидно, що якщо значення $\|M\|$ достатньо велике, то умова $\|\dot\sigma(\mathbf{z})\| \cdot \|M\| < 1$, взагалі кажучи, не може бути виконана і буде мати сенс тільки коли $\|\dot\sigma(\mathbf{z})\| \to 0$. Для $\sigma(cu) = tanh(cu)$, і тому $\dot\sigma(x) = d(\tanh(x))/dx = \operatorname{sech}^2(x) = 4/(\exp(x)+\exp(-x))^2$. Тоді умова

$$\|\dot\sigma(\mathbf{z})\| \cdot \|M\| = \frac{4\|M\|}{(\exp(\|\mathbf{z}\|)+\exp(-\|\mathbf{z}\|))^2} = \varepsilon^2 < 1 \tag{2.32}$$

має бути задоволена.

Введемо позначення $t = \exp(\|\mathbf{z}\|)$. Тоді нерівність (2.32) можна записати як

$$t^2 - 2\sqrt{\|M\|}/\varepsilon t + 1 > 0.$$

Звідси маємо:

$$t = \exp(\|\mathbf{z}\|) > (\sqrt{\|M\|} + \sqrt{\|M\| - \varepsilon^2})/\varepsilon.$$

Отже, якщо

$$\|\mathbf{z}\| > \ln \frac{\sqrt{\|M\|} + \sqrt{\|M\| - \epsilon^2}}{\epsilon} > 0,$$

то нерівність (2.32) буде задоволено. Дійсно, підставивши

$$\|\mathbf{z}\| = \ln \frac{\sqrt{\|M\|} + \sqrt{\|M\| - \epsilon^2}}{\epsilon} > 0, 0 < \epsilon < \sqrt{M},$$

то

$$\frac{2\epsilon\sqrt{\|M\|}(\sqrt{\|M\|} + \sqrt{\|M\| - \epsilon^2})}{(\sqrt{\|M\|} + \sqrt{\|M\| - \epsilon})^2 + \epsilon^2}$$
$$= \frac{2\epsilon\sqrt{\|M\|}(\sqrt{\|M\|} + \sqrt{\|M\| - \epsilon^2})}{2M + 2\sqrt{\|M\|}\sqrt{\|M\| - \epsilon^2}} = \epsilon < 1 < \sqrt{\|M\|}.$$

Ця нерівність означає, що $\epsilon^2 < 1$ і $\|M\| > 1$. Отже, для $\forall \|M\| > 1$ і $\forall \mathbf{u} \in \mathbb{S}$ також маємо $\|\dot{\sigma}(\mathbf{z})\| \cdot \|M\| < 1$ і $\mathbf{T}(\mathbf{u}) \in \mathbb{S}$. Використання теореми 2.3 завершує доведення для випадку $c = 1$.

Тепер розглянемо випадок $c < 1$. Очевидно, що в цьому випадку маємо $\dot{\sigma}(cu) = c \cdot d(\tanh(u))/du$, а отже умова $\|\dot{\sigma}(c\mathbf{z})\| \cdot \|M\| < 1$ виконується для $0 < c < 1$. □

Лема 2.2. *Нехай $\sigma(u) = w(u, \alpha, \beta, c)$. Якщо матриці A і W невироджені та $c < 1/\|M\|$, то система* (2.28) *має хоча б одну точку рівноваги.*

Доведення. Доведення Леми 2.2 повторює доведення Леми 2.1, якщо замінити підстановку $\sigma(cu) = \tanh(cu)$ на $\sigma(cu) = w(u, \alpha, \beta, c)$. У цьому випадку, з (2.22) випливає, що $\|\dot{\sigma}(x)\| < c$ і шар $\mathbb{S} := \{\mathbf{u} \in \mathbb{R}^n | \ \|\mathbf{u} - \mathbf{p}\|^2 \leq c\}$, де $\mathbf{p} = -M^{-1}\mathbf{g}$. Отже, якщо $\|M\| < 1/c$, то $\|\dot{\sigma}(\mathbf{z})\| \cdot \|M\| < 1$. □

Отже, система (2.28) має хоча б одну точку рівноваги $\mathbf{x}^* = -A^{-1}(\mathbf{u}^* + \mathbf{c})$, де $\mathbf{u}^*$ – розв'язок рівняння (2.29).

Нехай матриця W невироджена. Тоді введемо в системі $\mathbf{x}$ в (2.28) заміну змінної $\mathbf{y} = W\mathbf{x} + \mathbf{b}$:

$$\dot{\mathbf{y}}(t) = WAW^{-1}\mathbf{y}(t) - WAW^{-1}\mathbf{b} + W\mathbf{c} + W\sigma(\mathbf{y}(t)), \tag{2.33}$$

де $\sigma(u) = w(u, \alpha, \beta, c)$.

Позначимо WAW^{-1} як $H \in \mathbb{R}^{n \times n}$ і $-WAW^{-1}\mathbf{b} + W\mathbf{c}$ як $\mathbf{q} \in \mathbb{R}^n$. Це дозволить перейти до розгляду системи, еквівалентної (2.28):

$$\dot{\mathbf{y}}(t) = H\mathbf{y}(t) + \mathbf{q} + W\sigma(\mathbf{y}(t)). \tag{2.34}$$

Разом з поведінкою розв'язків системи (2.34), розглянемо також її одновимірний варіант:

$$\dot{y}(t) = hy(t) + q + p\sigma(y(t)), \tag{2.35}$$

де $p > 0$ і $h, q \in \mathbb{R}$.

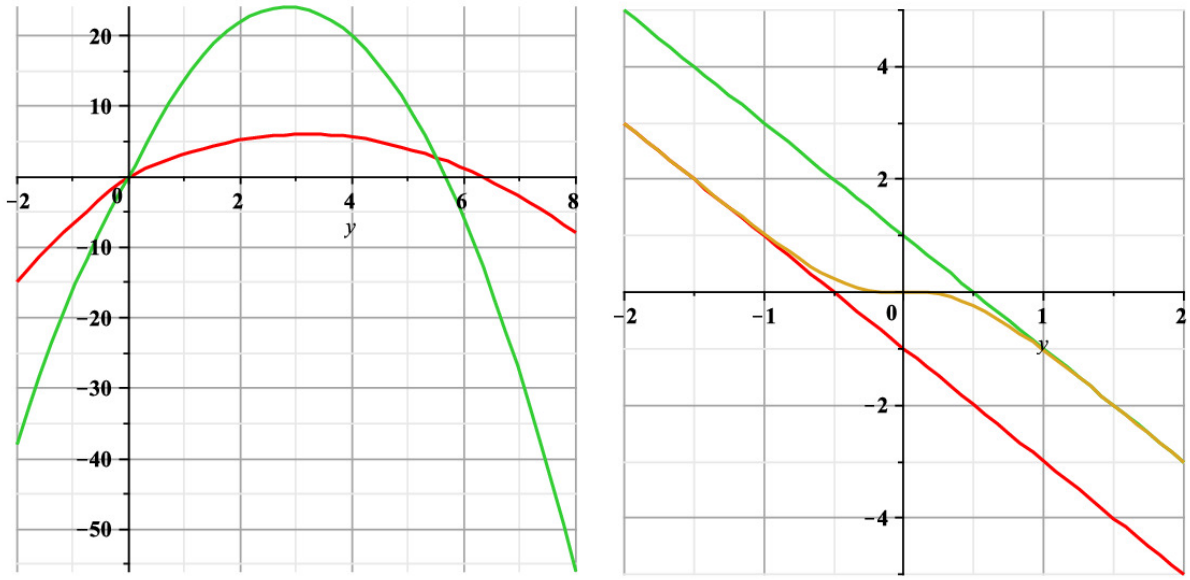
Випадок 1: $\sigma(y) = \tanh(cy)$

Представимо функцію $\tanh(cy)$ у вигляді її розкладання в ряд Тейлора:

$$\tanh(cy) = cy - \frac{(cy)^3}{3} + \frac{2(cy)^5}{15} + O(y^7). \tag{2.36}$$

Розглянемо функцію $V(y) = 0.5y^2$ та її похідну відносно (2.36):

$$\dot{V}_t(t) = hy^2 + qy + py \cdot \sigma(y). \tag{2.37}$$



(а) $\dot{V}_t(y) = hy^2 + by + py \cdot (\tanh(cy) - cy)$ (б) $r(y) = \tanh(cy) - cy$

Рис. 2.10: (а) функції $\dot{V}_t(y)$ для параметрів $h = 1, b = 15, p = 2, c = 2$ (червона) і $h = -3, b = 5, p = -1.2, c = 2$ (зелена); (b) та значення $r(y) \in [-cy - 1, -cy + 1]$ (між двома прямими лініями: $-cy-1$ (червона) і $-cy+1$(зелена) для параметра $c = 2$.

Зауважимо, що функції $\sigma(y)$ і cy непарні, а отже і функція $r(y) = \sigma(y) - cy$ також непарна і якщо $y > 0$, то $-cy - 1 < r(y) < 0$, а якщо $y < 0$, то $-cy + 1 > r(y) > 0$. Враховуючи ці зауваження, перепишемо функцію (2.37) як

$$\dot{V}_t(t) = [h + pc \cdot (1 - \gamma(y^2)]y^2 + (q \pm p)y$$

$$0 \leq \gamma(y^2) < 1.$$

За $t \to \infty$ маємо, що $\gamma(y^2) \to 1$, з чого випливає, що за умови $h < 0$

$$\lim_{t\to\infty} \dot{V}_t(t) \to -\infty.$$

Отже, якщо $h < 0$, то розв'язок $y(t)$ рівняння (2.35) є обмеженим. Цей висновок, а також те, що рівняння (2.35) має одну або три стійких точки, дозволить довести більш загальний результат.

Теорема 2.4. *Нехай в системі* (2.28) $\sigma(\boldsymbol{x}) = \tanh(\boldsymbol{x})$ *та матриця A Гурвіцева. Тоді будь-який розв'язок цієї системи обмежений.*

Доведення. Перепишемо (2.28) у вигляді інтегрального рівняння:

$$\mathbf{x}(t) = \exp(At)\mathbf{x}(0) + \int_0^t \exp(A(t-\tau))(\mathbf{c} + \sigma(W\mathbf{x}(\tau) + \mathbf{b})d\tau.$$

Звідси маємо:

$$\|\mathbf{x}(t)\| \le \|\exp(At)\|\|\mathbf{x}(0)\| + \int_0^t \|\exp(A(t-\tau))\|(\|\mathbf{c}\| + \|\sigma(W\mathbf{x}(\tau)+\mathbf{b})\|)d\tau. \quad (2.38)$$

Оскільки матриця A є Гурвіцевою, то при $\beta > 0$ справедлива нерівність $\|\exp(At)\| \le \beta\exp(-\alpha t)$, де $\alpha > 0$ – максимальна дійсна частина всіх власних чисел матриці A. Окрім цього, $-1 < \sigma(v) < 1$. Отже, маємо $\|\mathbf{c}\| + \|\sigma(W\mathbf{x}(\tau) + \mathbf{b})\| < \|\mathbf{c}\| + \sqrt{n}$ і можемо переписати нерівність 2.38 як:

$$\|\mathbf{x}(t)\| \le \beta\exp(-\alpha t)\|\mathbf{x}(0)\| + \beta(\|\mathbf{c}\| + \sqrt{n})\int_0^t \exp(-\alpha(t-\tau))d\tau$$

$$\le \beta\exp(-\alpha t)\|\mathbf{x}(0)\| + \frac{\beta(\|\mathbf{c}\| + \sqrt{n})}{\alpha}(1 - \exp(-\alpha t)).$$

Легко бачити, що за $t \to \infty$ для будь-якого розв'язку $\mathbf{x}(t)$ рівняння 2.28 буде справедлива оцінка $\|\mathbf{x}(t)\| \le \beta(\|\mathbf{c}\| + \sqrt{n})/\alpha$, що завершує доведення теореми 2.4. □

Теорема 2.5. *(ЛаСаля) [77] Нехай* $\mathbb{H} \subset \mathbb{R}^n$ *– компактна множина, що є додатно інваріантною відносно* (2.28)*. Нехай* $V : \mathbb{R}^n \to \mathbb{R}$ *– неперервно диференційовна функція, що задовольняє умові* $\dot{V}(\mathbf{x}) \le 0$ *або* $\dot{V}(\mathbf{x}) \ge 0$ *в* $\mathbb{H}$*. Нехай*

$\mathbb{E}$ – *множина усіх точок в* $\mathbb{H}$, *де* $\dot{V}(\mathbf{x}) = 0$. *Нехай* $\mathbb{M}$ – *найбільша інваріантна множина в* $\mathbb{E}$. *Тоді кожен розв'язок, що починається в* $\mathbb{H}$ *наближається до* $\mathbb{M}$ *при* $t \to \infty$.

Позначимо точки рівноваги системи (2.28) як $E_1 = (e_{11}, ..., e_{n1})^T, ..., E_k = (e_{1k}, ..., e_{nk}))^T$ $(k \geq 0)$ та знайдемо значення матриці Якобі в кожній з цих точок:

$$J(E_i) = A + \dot{\sigma}(E_i)W,$$

де

$$\dot{\sigma}(z) = \frac{d(\tanh(z))}{dz} = \mathrm{ch}^{-2}(z)$$
$$\dot{\sigma}(E_i) = \mathrm{diag}(\dot{\sigma}(w_{11}e_{1i} + \cdots + w_{1n}e_{ni}), ..., \dot{\sigma}(w_{n1}e_{1i} + \cdots + w_{nn}e_{ni})).$$

Теорема 2.6. *Нехай* $\{\lambda_{i1}, ..., \lambda_{in}\}$ – *власні числа матриці* $J(E_i)$, $i \in \{1, ..., k\}$. *Якщо система задовольняє умовам Теореми 2.1 і Лемі 2.1 і точки рівноваги* $E_i, i = 1, ..., k$ *нестабільні, то в системі існують непорожні інваріантні множини ненульової розмірності.*

Нехай існує хоча б одна точка $E_m, m \in \{1, ..., k\}$, *якій відповідає таке значення якобіана* $J(E_m)$, *що* $\forall j \in \{1, \ldots, n\}$, $\Re(\lambda_{mj}) < 0$ *і* $\exists l \in \{1, \ldots, n\}$, $\Im(\lambda_{ml}) \neq 0$. *Тоді в системі* (2.28) *існують граничні цикли, граничні тори, або хаотичні атрактори (для* $n > 2$*).*

Доведення. Доведення Теореми 2.6 ґрунтується на наступних трьох твердженнях.

1. Будь-який розв'язок системи (2.28) при $t \to \infty$ віддаляється від точок рівноваги.

2. Згідно з Теоремою 2.1 про обмеженість розв'язків, жоден з них не прямує до нескінченності.

3. Точка E_m – нестійкий фокус. Оскільки не існує стійких точок рівноваги, траєкторії, що починаються в околі E_m мають збігатися до граничної множини.

З наведених тверджень і Теореми 2.5 випливає, що існує обмежена множина, що містить деяку інваріантну підмножину ненульової розмірності (наприклад,

граничний цикл). За умовами Теореми 2.5, для системи (2.28) має існувати хоча б одна точка рівноваги, що гарантується Лемою 2.1. □

Розглянемо двовимірний та тривимірний варіант системи (2.28) з параметрами:

$$\begin{aligned} A &= \begin{pmatrix} 0.34 & 1.6 \\ -1.0 & -1.4 \end{pmatrix}, \ \mathbf{c} = \begin{pmatrix} 2 \\ 0 \end{pmatrix}, \\ W &= \begin{pmatrix} 1.8 & 1.0 \\ 0.0 & 1.6 \end{pmatrix}, \qquad \mathbf{b} = \begin{pmatrix} 0 \\ -1 \end{pmatrix}. \end{aligned} \tag{2.39}$$

$$\begin{aligned} A &= \begin{pmatrix} 0.34 & 1.6 \\ -1.0 & -0.4 \end{pmatrix}, \ \mathbf{c} = \begin{pmatrix} 0 \\ 0 \end{pmatrix}, \\ W &= \begin{pmatrix} 1.8 & 4.0 \\ 3.0 & 1.6 \end{pmatrix}, \qquad \mathbf{b} = \begin{pmatrix} 0 \\ 0 \end{pmatrix}. \end{aligned} \tag{2.40}$$

$$\begin{aligned} A &= \begin{pmatrix} -0.2 & 0.0 \\ a_{21} & 0.0 \end{pmatrix}, \ \mathbf{c} = \begin{pmatrix} 0 \\ 1.3 \end{pmatrix}, \\ W &= \begin{pmatrix} 1.6 & 4.0 \\ -4.0 & 1.8 \end{pmatrix}, \ \mathbf{b} = \begin{pmatrix} -10 \\ 10 \end{pmatrix}. \end{aligned} \tag{2.41}$$

$$\begin{aligned} A &= \begin{pmatrix} -0.2 & -0.11 & -0.22 \\ -2.0 & -0.10 & -1.0 \\ 5.9 & 3.9 & 0.0 \end{pmatrix}, \ \mathbf{c} = \begin{pmatrix} c_1 \\ 1.59 \\ 0.06 \end{pmatrix}, \\ W &= \begin{pmatrix} -5.0 & 4.0 & 4.0 \\ -4.0 & 1.8 & -1.0 \\ -10 & 10 & 8.0 \end{pmatrix}, \qquad \mathbf{b} = \begin{pmatrix} -10 \\ 20 \\ -10 \end{pmatrix}. \end{aligned} \tag{2.42}$$

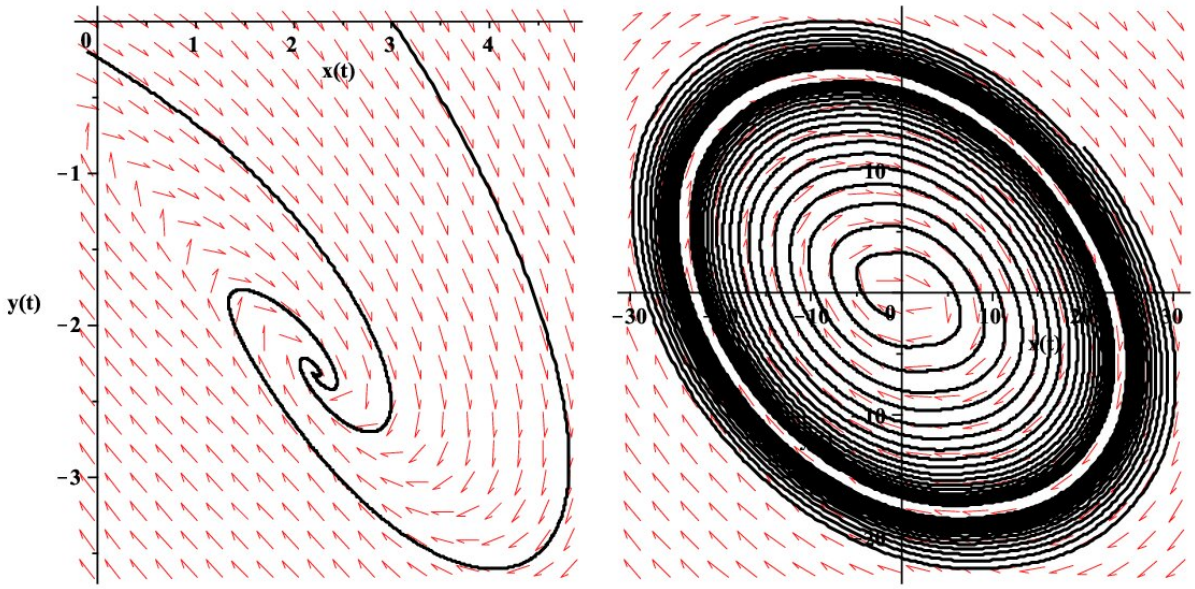
Приклади траєкторій системи (2.28) з наданими вище параметрами наведено на Рисунках 2.11, 2.12 та 2.13.

Випадок $\sigma(y) = w(y, \alpha, \beta, c)$

Розглянемо активаційну функцію (2.21).

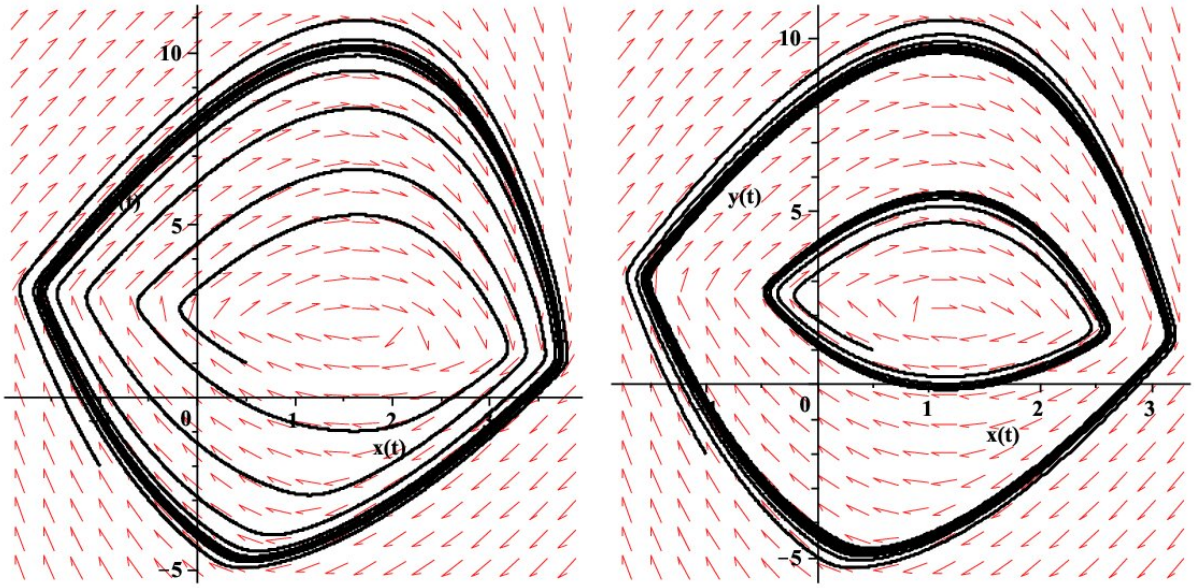
Теорема 2.7. *Нехай у системі* (2.34) $\sigma(y) = w(y, \alpha, \beta, c)$ *та симетрична матриця* $H + H^T$ *від'ємно визначена. Тоді у цій системі існує непорожня інваріантна множина.*

Доведення. Визначимо кандидата у функцію Ляпунова до системи (2.34) як



(а) $E_1 = (2.2348, -2.3105)$ (б) $E_1 = (0, 0)$

Рис. 2.11: Система (2.28) з параметрами (2.39) і (2.40), точка E_1 – нестійкий фокус, навколо якого існує стійкий граничний цикл.

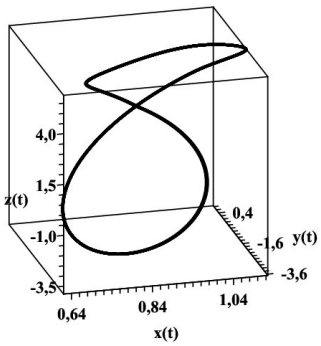


(а) $a_{21} = -1.4$ (б) $a_{21} = -2$

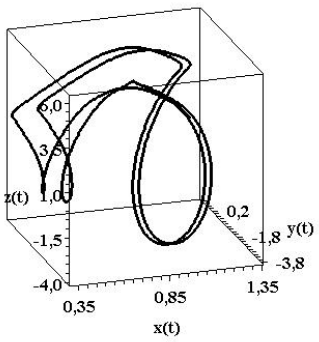
Рис. 2.12: Народження двох стійких граничних циклів з одного стабільного циклу в системі (2.28) в результаті варіювання параметра a_{11}; матриця A не є Гурвіцевою (існує нульове власне число; (a) один стійкий граничний цикл; (b) два стійких граничних цикла.

дійсну функцію $V(y_1, ..., y_n) = 0.5 \cdot (y_1^2 + ... + y_n^2)$. З такого визначення маємо:

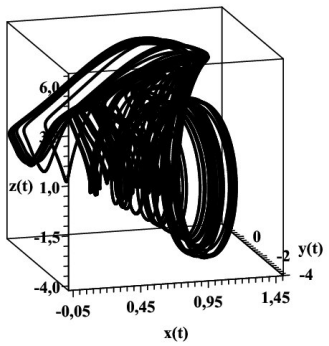
$$\dot{V}_t(y_1(t), ..., y_n(t)) = \mathbf{y}^T(H + H^T)\mathbf{y} + 0.5 \cdot \mathbf{y}^T W^T \sigma(\mathbf{y}) + 0.5 \cdot \sigma(\mathbf{y})^T W^T \mathbf{y}. \quad (2.43)$$



(а) $c_1 = 0.83$



(б) $c_1 = 0.93$



(в) $c_1 = 1.03$

Рис. 2.13: Народження хаотичного атрактора з граничного циклу в системі (2.28) з параметрами (2.42); біфуркації було досягнуто відносно параметра c_1: (a) 1-цикл, (b) 2-цикл, (c) хаотичний атрактор.

Оскільки $|\sigma(y)| \leq |y|$ при $|y| \to \infty$, то з останнього рівняння випливає нерівність:

$$\dot{V}_t(y_1(t), ..., y_n(t)) \leq \lambda_{\max} \cdot (y_1^2 + \cdots + y_n^2) + s(y_1, ..., y_n), \tag{2.44}$$

де $\lambda_{\max} < 0$ – найбільше власне число матриці $H + H^T$, $s(y_1, ..., y_n) = \sum \gamma_{i_1,...,i_n} y_i^{k_i} \cdot y_j^{k_j}$ – сума таких степеневих функцій, що $k_i + k_j \leq 2$, $i, j \in \{1, ..., n\}$.

Розв'язок нерівності (2.44) можна знайти за формулою:

$$V(y_1(t), ..., y_n(t)) \leq V_0 \cdot \exp \int_0^t \left[\frac{s(y_1(\tau), ..., y_n(\tau)) + \lambda_{\max} \cdot (y_1^2(\tau) + \cdots + y_n^2(\tau))}{V(y_1(\tau), ..., y_n(\tau))} \right] d\tau, \tag{2.45}$$

де константа $V_0 = V(y_1(0), ..., y_n(0)) > 0$.

Нехай $\mathbb{N} = \{-1 \leq u_1 \leq 1, ..., -1 \leq u_n \leq 1\}$ – гіперкуб зі стороною 2. Тоді при $c = 1$ та $\forall \mathbf{y} \in \mathbb{N}$ у системі (2.34) маємо $\sigma(\mathbf{y}) \in \mathbb{N}$.

Нехай степенева функція $\sigma(\mathbf{y})$ задовольняє умовам $\alpha < 1$ і $\beta < 1$, тоді існує $T_0 > 0$, що за $\mathbf{y}(t) \notin \mathbb{N}$ для $\forall t > T_0$ маємо $\sigma(y_1, ..., y_n) + \lambda_{\max} \cdot (y_1^2 + \cdots + y_n^2) < 0$ і

$$\lim_{t \to \infty} \frac{\sigma(y_1(\tau), ..., y_n(\tau)) + \lambda_{\max} \cdot (y_1^2(\tau) + \cdots + y_n^2(\tau))}{V(y_1(t), ..., y_n(t))} < 0. \tag{2.46}$$

Отже, отримуємо $V(y_1(t), ..., y_n(t)) \to 0$ за $t \to \infty$. Але з іншого боку, це означає, що за $\mathbf{y}(t) \in \mathbb{N}$ існує такий момент $T_1 > T_0 > 0$, що якщо $t > T_1$,

то $\sigma(y_1, ..., y_n) + \lambda_{\max} \cdot (y_1^2 + \cdots + y_n^2) > 0$ та додатна функція $V(y_1(t), ..., y_n(t))$ зростає.

Позначимо $\mathbb{G} = \{\mathbf{g}\} \subset \mathbb{R}^n$ як множину усіх точок, що задовольняють умові $\mathbf{g}^T(H + H^T)\mathbf{g} + 0.5 \cdot \mathbf{g}^T W^T \sigma(\mathbf{g}) + 0.5 \cdot \sigma(\mathbf{g})^T W^T \mathbf{g} \geq 0$. Легко бачити, що $\mathbb{G} \subset \mathbb{N}$.

Оскільки функція $\mathbf{g}^T(H + H^T)\mathbf{g}$ є негативно визначеною, то $\mathbb{G}$ компактна позитивно інваріантна множина відносно системи (2.34). Отже, якщо розв'язок системи (2.34) належить $\mathbb{G}$, то він обмежений, як і розв'язок $V(y_1, ..., y_n)$ рівняння (2.43).

Тепер припустимо, що існує такий момент T_u, що для $t = T_u$ $\mathbf{y}(T_u) \notin \mathbb{G}$. Тоді нерівність (2.46) справджується.

Позначимо як $\mathbb{S}$ шар з центром у початку координат, чия поверхня містить точку T_u. У цьому випадку $\mathbb{G} \subset \mathbb{S}$, а отже, в силу (2.46) та згідно з Теоремою 2.5, існує такий момент $T_s > T_u$, що $\mathbf{y}(T_s) \in \mathbb{G}$. Отримуємо, що розв'язок $\mathbf{y}(t)$ системи (2.34), що починається в $\mathbb{S}$, належить $\mathbb{S}$. До того ж $\mathbf{y}(t) \to \mathbb{G}$ при $t \to \infty$.

Застосуємо принцип порівняння [77] до розв'язку $V(\mathbf{y})$ рівняння (2.43) та розв'язку нерівності (2.44). Тоді отримаємо як наслідок обмеженість розв'язку $\mathbf{y}(t)$ системи (2.34). □

Теорема 2.8. *Нехай матриця A в системі* (2.28) *невироджена. Також нехай $E_i, i = 1, ..., k$ – точка рівноваги системи* (2.28) *та існує точка $E_m, m \in \{1, ..., k\}$, у якій матриця $A + \dot{\sigma}(E_m)W$ – Гурвіцева. Тоді існує окіл точки E_m, у якому усі розв'язки системи* (2.28) *обмежені.*

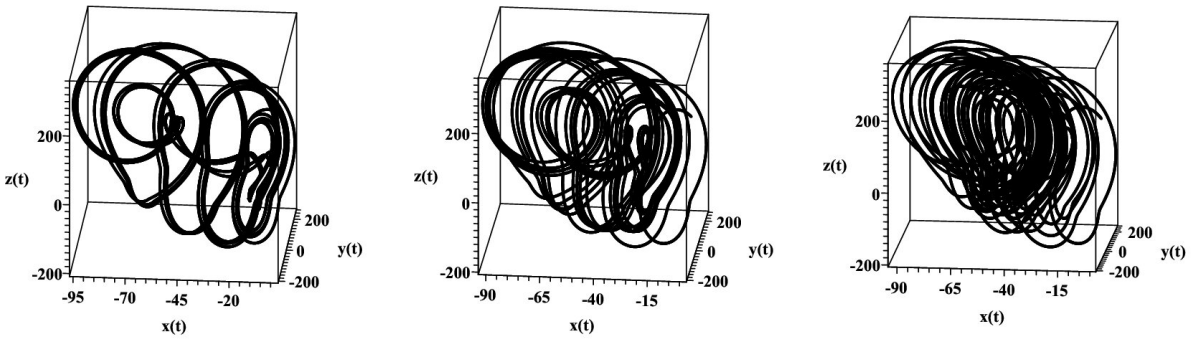
Наслідок 2.1. *Нехай у системі* (2.28) *вектори $\boldsymbol{b}$ і $\boldsymbol{c}$ нульові, а матриця $A + cW$ – Гурвіцева. Тоді існує окіл початку координат, у якому тривіальний розв'язок системи* (2.28) *асимптотично стійкий.*

Доведення. Згідно з Лемами 2.1 і 2.2, система (2.28) має хоча б одну точку рівноваги (позначимо її як E_m). Тепер застосуємо добре відому теорему Ляпунова про асимптотичну стійкість розв'язків систем диференціальних рівнянь [77]. □

Розглянемо тривимірну систему (2.28) з наступними параметрами:

$$A = \begin{pmatrix} -0.12 & -0.10 & -0.2 \\ -2.0 & -0.10 & -1.0 \\ 1.0 & 1.0 & 0.07 \end{pmatrix}, \quad \mathbf{c} = \begin{pmatrix} c_1 \\ 5.6 \\ -1.3 \end{pmatrix}, \\ W = \begin{pmatrix} -3.84 & 3.1 & 4.0 \\ -4.0 & 2.0 & -1.0 \\ -9.0 & 10 & 9.0 \end{pmatrix}, \quad \mathbf{b} = \begin{pmatrix} 0 \\ 2 \\ 1 \end{pmatrix}. \tag{2.47}$$

Поведінку траєкторій системи з такими параметрами наведено на Рис. 2.14



(а) $c_1 = -0.90$ (б) $c_1 = -0.95$ (в) $c_1 = -0.99$

Рис. 2.14: Народження хаотичного атрактора з граничного цикла у системі (2.28) з параметрами (2.47); біфуркації було досягнуто варіюванням параметра c_1: (a) 8-цикл, (b) 16-цикл, (c) хаотичний атрактор.

Розглянемо також параметри:

$$A = \begin{pmatrix} -0.12 & -0.11 & -0.22 \\ -2.0 & 0.0 & -1.0 \\ 5.9 & 3.9 & 0.0 \end{pmatrix}, \quad \mathbf{c} = \begin{pmatrix} -0.3 \\ 4.2 \\ 1.0 \end{pmatrix}, \\ W = \begin{pmatrix} -5.0 & -4.0 & 4.0 \\ 4.0 & 1.921 & -1.0 \\ -2.8 & 2.91 & 5.04 \end{pmatrix}, \quad \mathbf{b} = \begin{pmatrix} 0 \\ 0 \\ 6 \end{pmatrix}. \tag{2.48}$$

Траєкторії розв'язків системи (2.28) з параметрами (2.48) можна бачити на Рис.2.15.

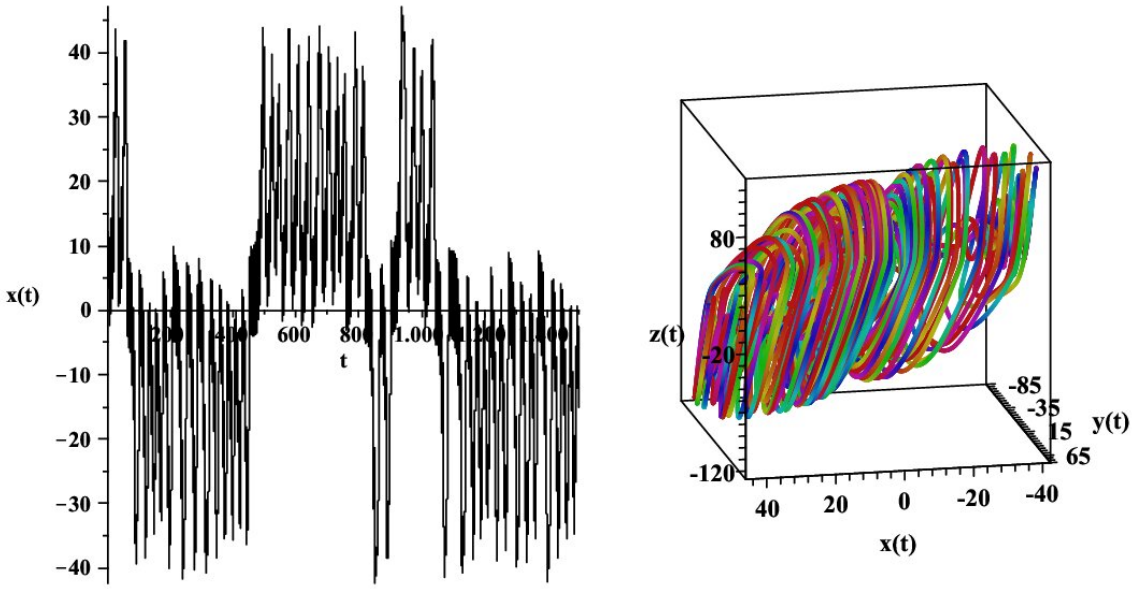


Рис. 2.15: Траєкторії системи (2.28) з параметрами (2.48)): (a) одна координата розв'язку $x(t)$, (b) хаотичний атрактор в $\mathbb{R}^3$.

2.6.2 Строге доведення існування хаосу в системі (2.28)

Нехай для системи (2.28) справджуються умови або Теореми 2.4, або Теореми 2.7, тоді легко перевірити, що для неї існує компактна додатно інваріантна множина $\mathbb{X}_0 \subset \mathbb{R}^n$.

Теорема 2.9. *Нехай початок координат O_e – сідлова точка рівноваги системи* (2.28)*, що належить інваріантній множині $\mathbb{X}_0$, а усі інші точки рівноваги в $\mathbb{X}_0$ нестійкі. Розглянемо вектор $(x_{10}, \ldots, x_{n0})^T \in \mathbb{X}_0$ початкових значень розв'язку $x_1, \ldots, x_n$, що задовольняє рівності:*

$$\liminf_{t\to\infty} V(t) = \liminf_{t\to\infty}(x_1^2(x_{10}, \ldots, x_{n0}, t) + \cdots + x_n^2(x_{10}, \ldots, x_{n0}, t)) = 0. \quad (2.49)$$

Тоді у системі (2.28) *присутня хаотична динаміка.*

Доведення. Для доведення теореми, застосуємо Теорему 3.1 з [55]. Спочатку, пояснимо побудову функції $V(\mathbf{x}) = \mathbf{x}^T\mathbf{x}$.

Зауважимо, що роль функцій $\mathbf{h}(\mathbf{x})$ і $\mathbf{g}(\mathbf{x})$ в Теоремі 3.1 тут виконують $\mathbf{x}$ та $\sigma(\mathbf{x})$, оскільки вони обидві є непарними та виконується нерівність $\mathbf{h}(\mathbf{x}) > \mathbf{g}(\mathbf{x})$.

Надалі покладемо $\mathbf{b} = \mathbf{c} = 0$.

Визначимо множину $\overline{\mathbb{X}}_0 \subset \mathbb{X}_0$ таку, що $\forall \mathbf{x}(t^*) \in \overline{\mathbb{X}}_0\ \dot{V}_t(t^*) = 2\mathbf{x}^T(t^*)\dot{\mathbf{x}}(t^*) = 0$; $t^* \in [0, \infty)$.

Оскільки матриця $A + A^T$ у Теоремі 2.4 (матриця $H + H^T$ у Теоремі 2.7) від'ємно визначена, то для достатньо великого значення $|\mathbf{x}|$ маємо:

$$\lim_{|\mathbf{x}|\to\infty} \dot{V}_t(\mathbf{x}) = \mathbf{x}^T(A + A^T)\mathbf{x} + w(\mathbf{x}) < 0,$$

де $|\mathbf{x}^T(A + A^T)\mathbf{x}| > |w(\mathbf{x})|$, $w(\mathbf{x}) = \sum \gamma_{i_1,\ldots,i_n} x_i^{k_i} \cdot x_j^{k_j}$ – сума таких степеневих функцій, що $k_i + k_j \leq 1 + \gamma$, $0 < \gamma < 1$, $i, j \in \{1, ..., n\}$.

Очевидно, що точка $\mathbb{O}_e$ має належати $\overline{\mathbb{X}}_0$.

Позначимо стійкі та нестійкі різноманіття точки $\mathbb{O}_e$ як $\mathbb{U}n \subset \mathbb{R}^n$ та $\mathbb{S}t \subset \mathbb{R}^n$ відповідно. Якщо траєкторія $\mathbf{x}(t) \subset \mathbb{S}t$, то вона наближається до точки $\mathbb{O}_e$, а якщо $\mathbf{x}(t) \subset \mathbb{S}t$, то віддаляється від неї. Оскільки точок рівноваги в $\mathbb{X}_0$ не існує, то траєкторія $\mathbf{x}(t)$ має прямувати до $\mathbb{O}_e$. Тоді, згідно з умовою (2.49), існує така послідовність моментів $t_1^* < t_2^* < \cdots < t_k^* < \ldots$, що при $t_k^* \to \infty$, $\forall \varepsilon > 0$, маємо $V(t_1^*) < \epsilon, \ldots, V(t_k^*) < \epsilon, \ldots$. Оскільки $V(t) > 0$ $(\forall t \in [0, \infty)\ V(t) \neq 0)$, то $\dot{V}_t(t_i^*) = 0, i = 1, \ldots, k, \ldots$.

Перепишемо умову (2.49) як

$$\liminf_{k\to\infty} V(t_k) = 0. \tag{2.50}$$

Також представимо нерівність для функції $V(t)$ з Теореми 2.44 у дещо зміненій формі:

$$\begin{aligned} &V(x_1(t), ..., x_n(t)) \\ &\leq V_0 + \int_0^t \Big[\lambda_{\max} \cdot (x_1^2(\tau) + \cdots + x_n^2(\tau)) + w(x_1(\tau), ..., x_n(\tau))\Big] d\tau, \end{aligned} \tag{2.51}$$

де константа $V_0 = V(x_1(0), ..., x_n(0)) > 0$ і $\lambda_{\max} < 0$ – найбільше власне значення матриці $A + A^T$ (або $H + H^T$).

Нехай в системі (2.28) має місце наступний випадок: зростаюча послідовність $t_k < t_{k+1} < t_{k+2} < \ldots$ відповідає максимумам функції $V(t)$.

Також нехай існує таке додатне число T, що $t_{k+1} - t_k = T$, $k = 1, 2, \ldots, \infty$. Використовуючи Першу Теорему про Середнє Значення для (2.51), отримуємо

наступний вираз:

$$\int_{t_k}^{t_k+T} \Big[\lambda_{\max} \cdot (x_1^2(\tau) + \cdots + x_n^2(\tau)) + w(x_1(\tau), ..., x_n(\tau))\Big] d\tau$$
$$\leq -\alpha_1(k) T V_k(\mu) + \beta_1(k) T V_k^{\gamma}(\xi), \tag{2.52}$$

де $\alpha_1(k) > 0$, $\beta_1(k) \in \mathbb{R}$, $t_k < \xi < t_{k+T}$, та $t_k < \mu < t_{k+T}$.

Далі, якщо $t \to \infty$, то значення $T > 0$ достатньо мале. У цьому випадку, можна припустити, що $\alpha_1(k)T = \alpha = const > 0$, $\beta_1(k)T = \beta = const \neq 0$, $V_k(\mu) = V_k(\xi) = V_k$.

Нехай $\beta > 0$. Тоді для послідовності $V_k = V(t_k), V_{k+1} = V(t_{k+1}), \ \ldots$ формула (2.51) може бути записана у вигляді:

$$V_{k+1} = V_k + \int_{t_k}^{t_{k+1}} \Big[\lambda_{\max} \cdot (x_1^2(\tau) + \cdots + x_n^2(\tau)) + w(x_1(\tau), ..., x_n(\tau))\Big] d\tau = -\alpha V_k + \beta V_k^{\gamma}. \tag{2.53}$$

де $\alpha = 1 - \alpha_1$, $T > 0$, $\beta = \beta_1 T > 0$ і $0 < \gamma < 1$. Для спрощення, позначимо $V_{k+1} = \Delta(V_k)$, $k = 0, 1, \ldots$, де $\Delta(V) \geq 0$ – одновимірне неперервне дійсне відображення задане (2.53).

Для продовження доведення Теореми 2.9, нам потрібні деякі результати про одновимірне відображення

$$\Delta(V) = \lambda \cdot (-\alpha V + \beta V^{\gamma}), \tag{2.54}$$

де $\alpha > 0$, $\beta > 0$, $\lambda > 0$ і $0 < \gamma < 1$.

Будемо називати будь-яку біфуркаційну діаграму, яка за структурою походить на діаграму логістичного відображення, діаграмою логістичного типу [78]—[83].

Лема 2.3. *Нехай існує інтервал $[\lambda_1, \lambda_2]$ варіювання параметра λ і таке значення $\beta_0 > 0$ параметра β, для яких існує біфуркаційна діаграма логістичного типу. Тоді структура біфуркаційної діаграми відображення* (2.54) *на інтервалі $[\lambda_1, \lambda_2]$ залишається незмінною для будь-якого значення $\beta \in (\beta_0, \infty)$.*

Доведення. Знайдемо регіон інваріантності L і нерухомі точки для відображе-

ння $\Delta(V) = \lambda \cdot (-\alpha V + \beta V^{\gamma})$, а також визначимо точки, які є притягуючими.

1. Регіон інваріантності L. Спершу визначимо границі цього регіону:

$$\Delta(V) = \lambda \cdot (-\alpha V + \beta V^{\gamma}) = 0 \Rightarrow V_{0i} = 0, V_{1i} = \Big(\frac{\beta}{\alpha}\Big)^{1/(1-\gamma)}.$$

Визначимо максимальне значення функції $\Delta(V)$ всередині сегмента $[V_{0i}, V_{1i}]$:

$$\Delta^{'}(V) = \lambda \cdot (-\alpha + \beta\gamma V^{\gamma-1}) = 0 \Rightarrow V_{\max} = \Big(\frac{\beta\gamma}{\alpha}\Big)^{1/(1-\gamma)} < V_{1i},$$

$$\Delta_{\max}(V_{\max}) = \lambda\Big[-\alpha\Big(\frac{\beta\gamma}{\alpha}\Big)^{1/(1-\gamma)} + \beta\Big(\frac{\beta\gamma}{\alpha}\Big)^{\gamma/(1-\gamma)}\Big] =$$

$$\lambda\alpha\Big(\frac{\beta\gamma}{\alpha}\Big)^{1/(1-\gamma)}\Big[-1 + \frac{\beta}{\alpha}\Big(\frac{\beta\gamma}{\alpha}\Big)^{-1}\Big] = \lambda\alpha\Big(\frac{\beta\gamma}{\alpha}\Big)^{1/(1-\gamma)}\Big[-1 + \frac{1}{\gamma}\Big].$$

Тоді має задовольнятися наступна умова:

$$\lambda\alpha\Big(\frac{\beta\gamma}{\alpha}\Big)^{1/(1-\gamma)}\Big[-1 + \frac{1}{\gamma}\Big] \leq \Big(\frac{\beta}{\alpha}\Big)^{1/(1-\gamma)}.$$

Звідси маємо:

$$\lambda\alpha(1-\gamma)\gamma^{\gamma/(1-\gamma)} \leq 1. \tag{2.55}$$

Отже, λ не залежить від β і з (2.55) маємо:

$$L = \left[0, \lambda\alpha\Big(\frac{\beta\gamma}{\alpha}\Big)^{1/(1-\gamma)}\Big[-1 + \frac{1}{\gamma}\Big]\right] \subset \left[0, \Big(\frac{\beta}{\alpha}\Big)^{1/(1-\gamma)}\right] \text{ and } \Delta(L) \subset L.$$

2. Нерухомі точки. Маємо:

$$V = \lambda \cdot (-\alpha V + \beta V^{\gamma}) \Rightarrow \frac{1}{\lambda} + \alpha = \frac{\beta}{V^{1-\gamma}}.$$

Отже, для фіксованих параметрів λ^* і α^* усі нерухомі точки V^* відображення $\Delta(V)$ на біфуркаційній діаграмі $V = f(\lambda)$ можна розмістити на одній вертикальній лінії $\lambda = \lambda^*$. Конкретні координати цих точок обчислюються за формулами:

$$V^{\gamma-1} = \frac{1}{\lambda\beta} + \frac{\alpha\lambda}{\lambda\beta}; V = \Big(\frac{\lambda\beta}{1+\alpha\lambda}\Big)^{1/(1-\gamma)}.$$

Отже, $V_0^* = 0$ і $V_1^* = \left(\frac{\lambda\beta}{1+\alpha\lambda}\right)^{1/(1-\gamma)}$. Зауважимо, що $V_1^* = \left(\frac{\lambda\beta}{1+\alpha\lambda}\right)^{1/(1-\gamma)} < \left(\frac{\beta}{\alpha}\right)^{1/(1-\gamma)} = V_{1i}$.

Тож $V_0^* \in L$ і $V_1^* \in L$.

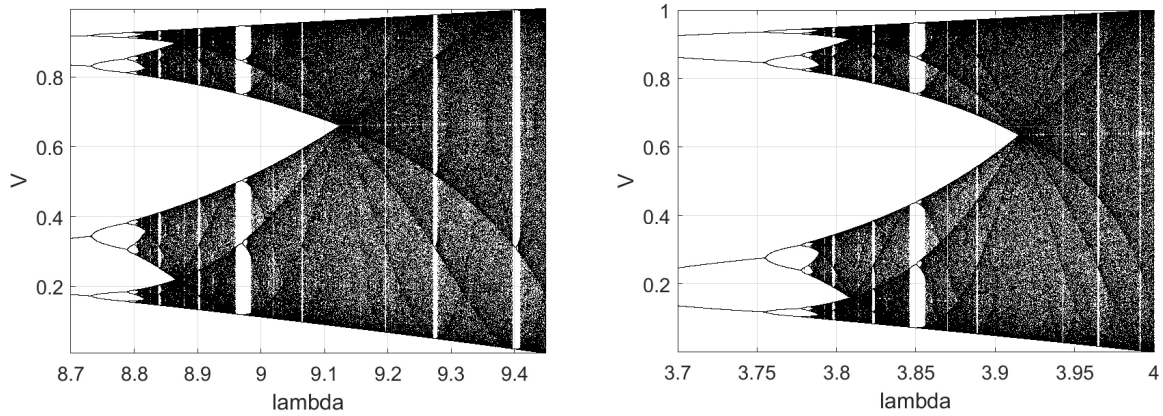
3. Притягуючі точки. Відомо, що з неперервною зміною параметра λ притягуючі нерухомі точки стають відштовхуючими й навпаки. Також відомо, що для того, щоб точка V^* була притягуючою, достатньо, щоб $|\Delta'(V^*)| < 1$. Отже, маємо $\Delta'(V_0^* = 0) = \alpha\lambda < 1$ $(\alpha > 0)$ і

$$|\Delta'(V_1^*)| = \left|\lambda\left[-alpha + \beta\gamma\left(\frac{\lambda\beta}{1+\alpha\lambda}\right)^{(\gamma-1)/(1-\gamma)}\right]\right| = |-\alpha\lambda(1-\gamma)+\gamma| < 1.$$

З відношення (2.55) випливає, що

$$\alpha\lambda(1-\gamma) < \gamma < 1 \Rightarrow \alpha\lambda < \frac{\gamma}{1-\gamma}\ (\alpha > 0).$$

Отже, починаючи з деякого $\beta_0 > 0$ та за умови сталості решти параметрів, структура біфуркаційної діаграми для $\forall\beta > \beta_0$ залишається такою ж, як і за $\beta = \beta_0$. Приклади діаграм наведено на Рисунках 2.16 та 2.17.

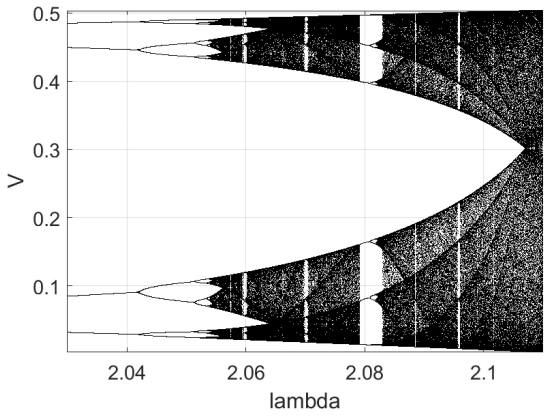


(а) $\alpha = -1.0, \beta = 1.0, \gamma = 0.75$ (б) $\alpha = -1.0, \beta = 1.0, \gamma = 0.5$

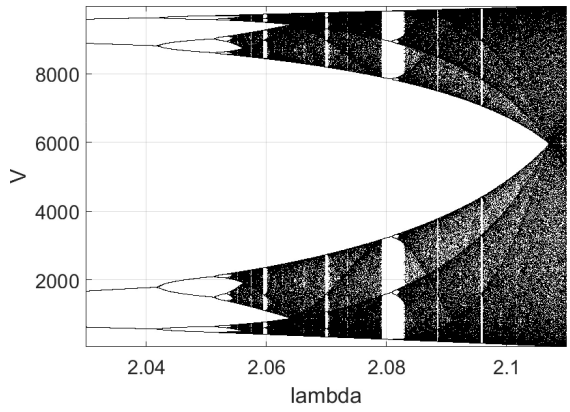
Рис. 2.16: Біфуркаційна діаграма одновимірного відображення $\Delta(V) = \lambda \cdot (-\alpha V + \beta V^\gamma)$ відносно значень параметра λ: (а) $\lambda \in [8.7-9.45]$; (b) $\lambda \in [3.7-4.0]$

□

Тепер повернемось до доведення Теореми 2.9. Головна ідея доведення в тому, що параметри дискретного процесу $V_{k+1} = \Delta(V_k) = \lambda\cdot(-\alpha V_k + \beta V_k^\gamma)$ завжди можна обрати так, щоб система (2.28) мала хаотичну поведінку. Це досягається



(а) $\alpha = -1.0, \beta = 0.6, \gamma = 0.25$



(б) $\alpha = -1.0, \beta = 10000, \gamma = 0.25$

Рис. 2.17: Біфуркаційна діаграма одновимірного відображення $\Delta(V) = \lambda \cdot (-\alpha V + \beta V^{\gamma})$ для ідентичних значень параметра λ: (a), (b) $\lambda \in [2.03 - 2.11]$. Діаграма зберігає свою структуру, а також вікна періодичної та хаотичної поведінки для усіх значень параметра $\beta \in [0.6, \infty)$.

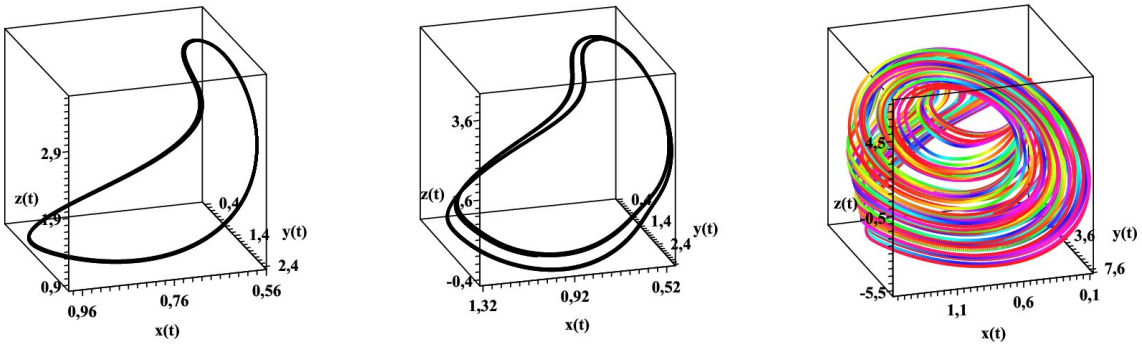
звичайними методами, які використовуються для доведення хаотичної поведінки відображення $V_{k+1} = \lambda V_k(1 - V_k)$ [79], [83]. За деяких значень параметрів процес (2.54) має хаотичну поведінку. Отже, існують значення параметрів, за яких розв'язки системи (2.28) також будуть хаотичними. □

Теорему 2.9 можна перевірити на практичних прикладах. Розглянемо тривимірну систему (2.28) з параметрами

$$A = \begin{pmatrix} -0.2 & -0.11 & -0.22 \\ -2.0 & 0.12 & 1.00 \\ a_{31} & -1.2 & -0.2 \end{pmatrix}, \; W = \begin{pmatrix} -5.0 & 4.0 & 1.0 \\ -4.0 & 1.8 & -4.0 \\ -1 & -2 & 1.0 \end{pmatrix}, \qquad (2.56)$$
$$\mathbf{b} = \mathbf{c} = 0$$

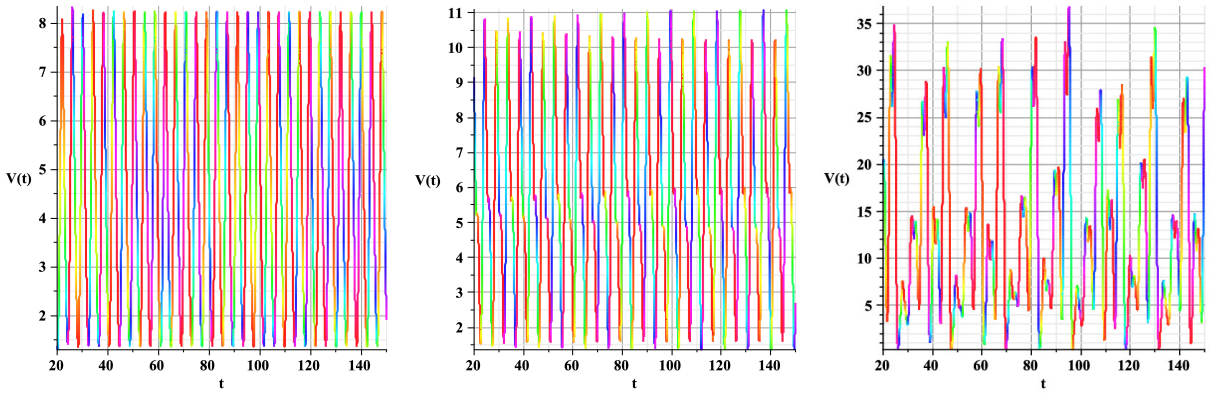
де приклади траєкторій наведено на Рисунках 2.18 та 2.19.

Також розглянемо приклад, коли умови Теореми 2.9 не виконуються (див. Рис. 2.20). Застосування Теореми 2.9 у цьому випадку демонструє, що деякі обмеження в її формулюванні можна опустити – початок координат не є точкою рівноваги, але множина $\mathbb{X}_0$ все ще містить сідлову точку.



(а) $a_{31} = 3.5$ (б) $a_{31} = 4.26$ (в) $a_{31} = 6.5$

Рис. 2.18: Народження хаотичного атрактора в системі (2.28) з параметрами (2.56) з граничного циклу; біфуркації було досягнуто відносно параметра a_{31}: (a) 1-цикл, (b) 2-цикл, (c) хаотичний атрактор. Початок координат $\mathbb{O}_e$ – сідлофокус; власні значення матриці A у випадку (c) дорівнюють: $-0.5023, 0.1111 + 1.5816i, 0.1111 - 1.5816i$.



(а) $a_{31} = 3.5$ (б) $a_{31} = 4.26$ (в) $a_{31} = 6.5$

Рис. 2.19: Підтвердження умов Теореми 2.9 про існування хаотичного атрактора в системі (2.28) з параметрами (2.56): (a) $\liminf_{t\to\infty} V(x_0, y_0, z_0, t) = 1.25$, (b) $\liminf_{t\to\infty} V(x_0, y_0, z_0, t) = 1.20$, (c) $\liminf_{t\to\infty} V(x_0, y_0, z_0, t) = 0.0$;.

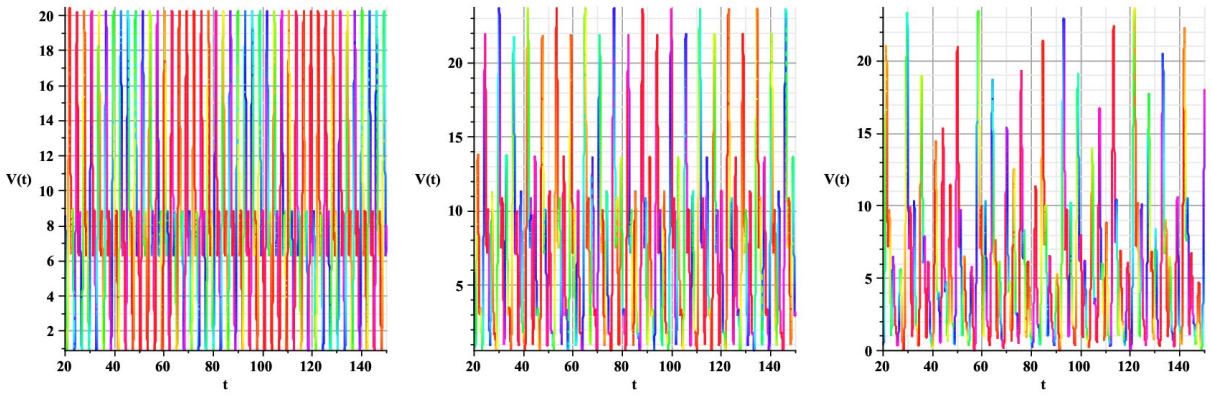
2.6.3 Стосовно системи (2.28) без лінійної частини

Зазвичай моделі на базі NODE будуються та розглядається у наступному вигляді:

$$\dot{\mathbf{x}}(t) = \sigma(W\mathbf{x}(t) + \mathbf{b}), \tag{2.57}$$

де матриця $W \in \mathbb{R}^{n\times n}$, вектор $\mathbf{b} \in \mathbb{R}^n$, і $\sigma(u)$ – скалярна активаційна функція.

Перепишемо систему (2.53) у наступному вигляді, замінивши $\sigma(u)$ вектор-



(а) $c_1 = 0.83$ (б) $c_1 = 0.93$ (в) $c_1 = 1.03$

Рис. 2.20: Підтвердження умов Теореми 2.9 про існування хаотичного атрактора в системі (2.28) з параметрами (2.42): (a) $\liminf_{t\to\infty} V(x_0, y_0, z_0, t) = 0.9$, (b) $\liminf_{t\to\infty} V(x_0, y_0, z_0, t) = 0.4$, (c) $\liminf_{t\to\infty} V(x_0, y_0, z_0, t) = 0.0$.

ною функцією:

$$\sigma(W\mathbf{x}(t) + \mathbf{b}) = \begin{pmatrix} \sigma_1(w_{11}x_1(t) + \cdots + w_{1n}x_n(t) + b_1), \\ \cdots \\ \sigma_n(w_{n1}x_1(t) + \cdots + w_{nn}x_n(t) + b_n) \end{pmatrix},$$

де $\sigma(u_1, ..., u_n) = (\sigma_1(u_1), ..., \sigma_n(u_n))^T \in \mathbb{R}^n$.

Визначимо $\sigma_i(u_i)$, $i = 1, \ldots, n$ наступним чином:

$$\sigma_i(u_i) = f_i(u_i) + \sum_{j=1}^{k} c_j f_{ij}(u_i) = \phi_i(u_i) \vee \psi_i(u_i) + \sum_{j=1}^{k} c_{ij}\phi_{ij}(u_i) \vee \psi_{ij}(u_i), \quad (2.58)$$

де $f_i(u_i)$ та $f_{ij}(u_i)$ – функції активації типу (2.11) або (2.12), а $c_{ij} \in \mathbb{R}^n$.

Беручи до уваги (2.58), введемо наступні функції:

$$F_i(u) := \int_0^u (\phi_i(\omega) \vee \psi_i(\omega))d\omega + \int_0^u \sum_{j=1}^{k} c_{ij} \cdot (\phi_{ij}(\omega) \vee \psi_{ij}(\omega))d\omega; i = 1, ..., n$$

$$G_i(u) := \int_{-u}^0 (\phi_i(\omega) \vee \psi_i(\omega))d\omega + \int_{-u}^0 \sum_{j=1}^{k} c_{ij} \cdot (\phi_{ij}(\omega) \vee \psi_{ij}(\omega))d\omega; i = 1, ..., n.$$

Теорема 2.10. *Нехай для системи* (2.57) *виконуються умови:*

1. $\det W \neq 0$;

2. $W + W^T$ *– від'ємно визначена;*

3. $f_i(u_i)$ *парна для будь-яких* i*;*

4. $\forall i \in \{1, ..., n\}$ $\lim\limits_{u\to\infty} F_i(u) > 0$ *і* $\lim\limits_{u\to-\infty} G_i(u) > 0$.

Тоді існує непорожня компактна інваріантна множина.

Доведення. Раніше було введено заміну змінної $\mathbf{y} = W\mathbf{x} + \mathbf{b}$ для переходу від системи (2.28) до (2.34). Оскільки $\det W \neq 0$, то так само можемо перейти від (2.57) до системи

$$\dot{\mathbf{y}}(t) = W\sigma(\mathbf{y}(t)), \mathbf{y}(0) = \mathbf{y}_0. \tag{2.59}$$

Тепер, використовуючи систему (2.59), введемо функцію

$$\dot{V}_t(\mathbf{y}) = \sigma^T(\mathbf{y})\dot{\mathbf{y}} + \dot{\mathbf{y}}^T\sigma(\mathbf{y}) = \sigma^T(\mathbf{y})(W + W^T)\sigma(\mathbf{y}).$$

Тоді з умови 2 Теореми 2.10 випливає, що $\dot{V}_t(\mathbf{y}) \leq 0$.

Тепер визначимо таку множину $\mathbb{L} \subset \mathbb{R}^n$, що для $\forall \mathbf{y} \in \mathbb{L}$ будемо мати $\dot{V}_t(\mathbf{y}) = 0$. З умов 3 та 4 Теореми 2.10 отримуємо:

$$\begin{aligned} V(\mathbf{y}) &= \int_0^\infty (\sigma^T(\mathbf{y})\dot{\mathbf{y}} + \dot{\mathbf{y}}^T\sigma(\mathbf{y}))dt = \sum_{i=1}^n F_i(u) \\ &= \sum_{i=1}^n \int_0^{y_i} (\phi_i(\omega) \vee \psi_i(\omega))d\omega + \Phi(y_1, ..., y_n) = C(\mathbf{y}_0) = const > 0, \end{aligned}$$

де $\Phi(y_1, ..., y_n)$ – неперервна функція. Тут спираємося на той факт, що інтеграл парної функції є непарною функцією [54], [55]. Таким же чином отримуємо:

$$\begin{aligned} V(\mathbf{y}) &= \int_{-\infty}^0 (\sigma^T(\mathbf{y})\dot{\mathbf{y}} + \dot{\mathbf{y}}^T\sigma(\mathbf{y}))dt = \sum_{i=1}^n G_i(u) \\ &= \sum_{i=1}^n \int_{-y_i}^0 (\phi_i(\omega) \vee \psi_i(\omega))d\omega + \Psi(y_1, ..., y_n) = C(\mathbf{y}_0) = const > 0, \end{aligned}$$

де $\Psi(y_1, ..., y_n)$ – неперервна функція. Отже, з умови 4 маємо:

$$\lim_{t\to\infty} V(\mathbf{y}(t)) > 0.$$

Зауважимо, що якщо значення $t = t_f < \infty$ скінченне, то $V(\mathbf{y}(t_f)) < 0$ виключити не можна.

Це означає, що множина $\mathbb{L}$ компактна та інваріантна для системи (2.59). Отже, множина $\mathbb{D} := \{\mathbf{y}(t) | \dot{V}_t(\mathbf{y}) \leq 0\}$ має границю $\mathbb{L}$ і також є компактною для цієї системи [77]. Якщо зробити обернену заміну змінної $\mathbf{x} = W^{-1}\mathbf{y} - W^{-1}\mathbf{b}$, то отримаємо такий же вираз, як і для системи (2.57). □

Отже, Теорема 2.10 дозволяє значно розширити клас активаційних функцій, які можна використовувати для побудови нейромережевих архітектур.

2.6.4 Порівняння нейронних мереж з різними активаційними функціями

Умови, наведені в Теоремах 2.4, 2.6-2.8, і 2.9, дозволяють будувати рекурентні нейронні мережі на основі системи (2.28), надаючи гарантії того, що її розв'язки будуть обмеженими. Цей факт є вирішальним при моделюванні будь-яких динамічних процесів. У цьому підрозділі розглянемо деякі такі моделі.

Конкретніше, розглянемо різні конфігурації та активаційні функції в контексті задачі апроксимації гармонічних сигналів, семплованих з частотою 16 кГц. Частота модельованого сигналу варіюється в межах від 60 до 600 герців і кожен з них може містити до 3-х гармонік.

Визначення моделі hNODE робить її привабливим кандидатом для моделювання подібних даних. Непрямий зв'язок вхідних та вихідних даних моделі дозволяє визначати функцію втрат, яка має на меті оптимізацію якісних характеристик побудованої системи, а не лише квадрату відхилення прогнозу від цільових даних. Іншими словами, інтегрування прихованого стану моделі може продовжуватися скільки завгодно довго без необхідності перевизначати його за допомогою моделі-кодувальника, а у контексті моделювання шумних або хаотичних даних це означає, що кодувальник має отримати лише початкові умови, після чого інтегрування, параметризоване безпосередньо з модельованих даних, отримає траєкторію достатньої довжини для отримання якісних характеристик системи та їх порівняння з бажаними значеннями. Практичним прикладом такого підходу є моделювання шумних періодичних сигналів – модель може коректно ідентифікувати такий сигнал, але згенерувати траєкторію, що зсунута за фазою відносно цільових значень. Наївна функція втрат некоректно іден-

тифікує такий результат як неправильний, хоча спектральні характеристики згенерованого сигналу будуть повністю відповідати очікуваним.

Ще однією перевагою hNODE у контексті поточної задачі є те, що крок інтегрування вкладеної системи ЗДР можна варіювати залежно від частоти дискретизації вхідних даних – якщо частота дискретизації висока, то крок інтегрування пропорційно зменшується, і навпаки. Це дозволяє моделі генерувати однаково коректні результати у різних середовищах.

Далі наведено процес побудови моделі та огляд окремих її компонент.

Перше, що треба вирішити при створенні моделі з архітектурою hNODE – це розмірність d її прихованого стану або латентного простору, у якому відбувається еволюція траєкторій вкладеної системи ЗДР. Для даної задачі, розмірність було визначено шляхом обчислення вкладення методом PECUZAL [56] конкатенації семплів з датасета LibriSpeech [84]. В результаті було отримано оптимальну розмірність простору вкладення $d = 10$. Такої розмірності має вистачити для вкладення простих гармонічних сигналів.

Друга задача – визначення структури вкладеної системи ЗДР. Як згадувалося раніше, головна мета даного чисельного експерименту полягає у порівнянні якості прогнозів, згенерованих моделями з різними активаційними функціями та їх комбінаціями. Як основу вкладеної системи було обрано звичайну одношарову щільну нейромережу, а для представлення комбінованих лінійної та нелінійної функцій активації було обрано лінійну комбінацію двох одношарових нейромереж.

У випадку лінійних систем, активаційна функція $\sigma(u) = u$ – тотожне відображення, яке не приносить в систему жодних нелінійних зв'язків. Рівняння такої системи ЗДР має вигляд (2.5).

У випадку нелінійних систем, було обрано степеневу активаційну функцію (2.21), оскільки було продемонстровано її ефективність у подібних задачах моделювання у порівнянні з, наприклад, гіперболічним тангенсом або ReLU. Параметри α, β, і c не фіксувалися і виводилися моделлю разом зі значеннями матриці W і вектора $\mathbf{b}$. Система ЗДР у цьому випадку задається у вигляді (0.1).

Комбінована нейромережа, яка місить одночасно лінійні та нелінійні компоненти, має вигляд (2.7), де функція $\sigma(u)$ – степенева функція активації з нефіксованими параметрами.

Параметри вкладеної NODE моделі визначаються контролером з поточного

контексту. Контекст складається з 266 семплів вхідних даних, а контролер – окрема нейромережа, побудована на основі LSTM блока [68]. LSTM блок – це функція з $\mathbb{R}$ в $\mathbb{R}^{128}$, що рекурсивно застосовується до кожного вхідного семпла для побудови вектора зі 128 елементами. Цей вектор потім подається на вхід нелінійної нейромережі, що використовує гіперболічний тангенс у якості функції активації, та перетворюється на вектор з $\mathbb{R}^{256}$. Результат подається на вхід лінійній моделі, що перетворює його на вектор з $\mathbb{R}^M$, де M – кількість параметрів вкладеної NODE моделі. Для лінійної моделі, що використовує тотожне відображення як функцію активації, $M = 110$. Для нелінійної моделі, що містить додаткові 3 параметри для степеневої функції активації, $M = 113$. Нарешті, для комбінованої моделі, що у своєму векторі параметрів містить дві матриці, два вектори, а також параметри для функції активації, $M = 223$.

Початкові умови для вкладеної системи ЗДР визначаються за допомогою ще однієї нейромережевої моделі – кодувальника. Кодувальник складається з двох шарів: перший перетворює вхідний сегмент з $\mathbb{R}^{266}$ на вектор в $\mathbb{R}^{532}$, використовуючи гіперболічний тангенс як функцію активації; а другий перетворює вектор з $\mathbb{R}^{532}$ на вектор з $\mathbb{R}^d$.

Нарешті, модель-декодер, яка перетворює траєкторії системи ЗДР в просторі $\mathbb{R}^d$ на одновимірний часовий ряд. Декодер побудовано у вигляді двошарового персептрона, що перетворює кожну точку з $\mathbb{R}^d$ на вектор з $\mathbb{R}^{2d}$ і потім в цільовий простір $\mathbb{R}$.

Функція втрат, яка використовувалася для тренування кожної з моделей, брала до уваги як спектральні характеристики згенерованих траєкторій, так і їх безпосереднє відхилення від бажаних значень. Спектр обчислювався шляхом застосування вікна Ганна до вхідного сигналу та до згенерованої траєкторії, побудови для отриманих сегментів дискретного перетворення Фур'є, та обчислення логарифмів абсолютних значень першої половини отриманих перетворень, після чого обчислювався квадрат відхилення спектрів цільового та згенерованого сигналів. Включення до функції втрат квадрата відхилення безпосередньо згенерованих значень від цільових забезпечує коректне тренування кодувальника та декодувальника і дозволяє моделі зберігати інформацію про фазу модельованого сигналу. А порівняння спектрів допомагає моделі забезпечити включення правильних гармонічних компонент у згенерованому сигналі.

Тренування усіх моделей тривало 256 епох, причому тренувальні дані гене-

рувалися наново кожні 16 епох. Графіки динамік втрат на тестових даних для кожної моделі наведено на Рис. 2.21

На графіках можна побачити деякі спільні риси, такі як повільний зріст значень функції втрат до $\sim$ 240-ї епохи, після чого досягається пік та значення починають спадати. Значення функції втрат для усіх моделей виявилися високими, проте усі моделі показали адекватний рівень короткострокової апроксимації, що продемонстровано на Рис. 2.22-2.24.

Варто зауважити, що тренування лінійної моделі забирає найменше часу – усі 256 епох було пройдено за 249 хвилин. Модель, що використовує степеневу функцію активації, тренувалась 3119 хвилин. А комбінована модель – 4320 хвилин. Така разюча різниця в часі тренування нелінійних моделей пояснюється тим, що параметри α, β, і c також мали обиратися моделлю. Деякі комбінації значень цих параметрів значно підвищують обчислювальні ресурси, які необхідні для застосування таких функцій. Якщо ці значення зробити фіксованими, наприклад, $\alpha = 1/2$, $\beta = 1/3$, $c = 1$, то вартість обчислення значно зменшиться, оскільки для квадратного та кубічного кореня будуть використовуватися спеціалізовані високоефективні алгоритми.

Як згадувалося раніше, hNODE дозволяє будувати пояснювані моделі та досліджувати їх властивості методами, розробленими у традиційних математичних дисциплінах, наприклад, теорії стійкості, хаотичній динаміці, або теорії біфуркацій. Для усіх побудованих вище моделей можна обчислити власні значення матриць, породжених моделлю-контролером, щоб визначити їх зв'язок з поведінкою модельованих даних та отримати уявлення про стійкість траєкторій вкладеної NODE моделі. Наприклад, матриці A і W для моделі (2.28), що апроксимує періодичний сигнал з частотою 330 Гц і трьома гармоніками, мають

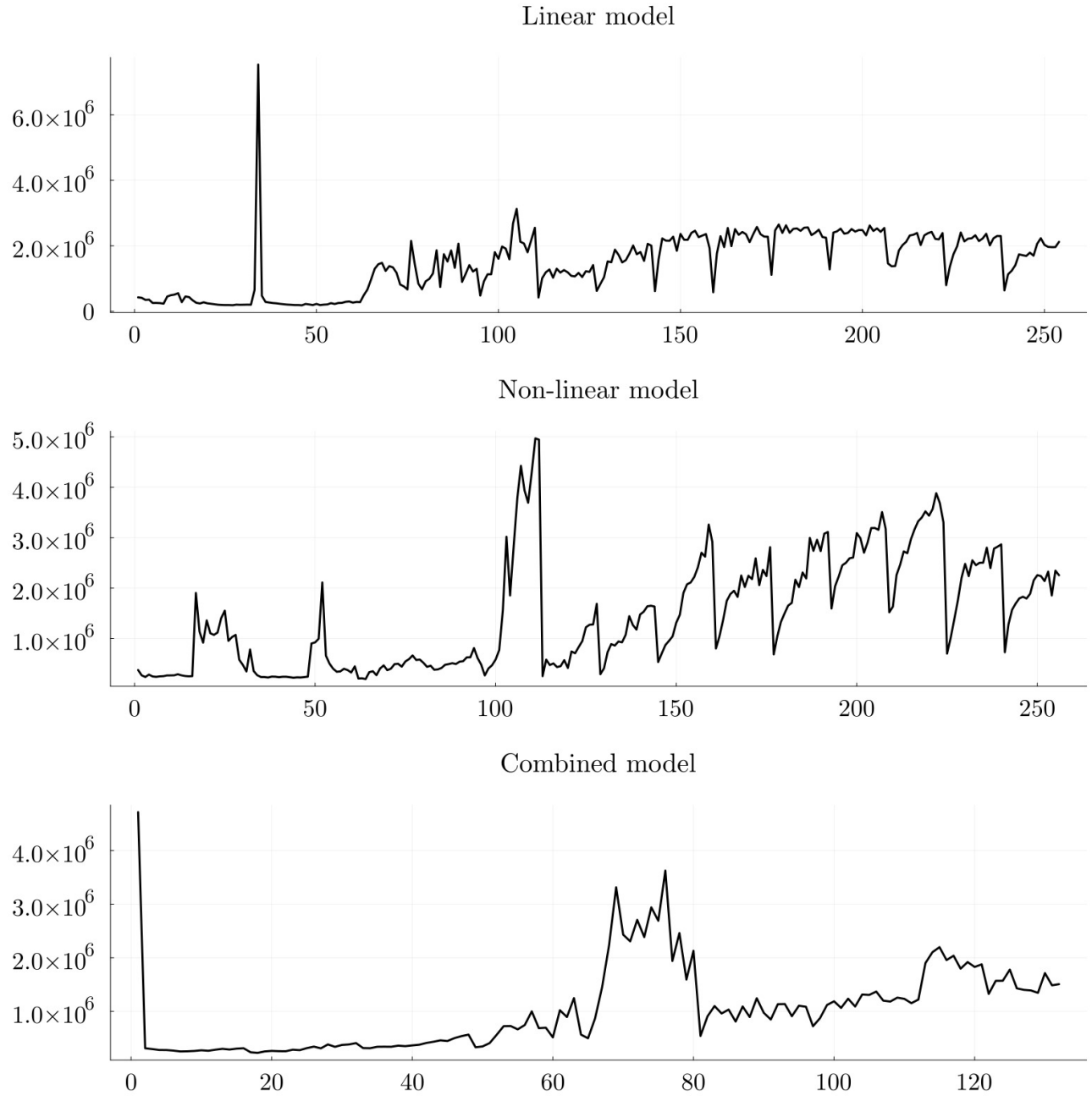


Рис. 2.21: Графіки втрат з плином епох для тьох типів вкладених NODE моделей.

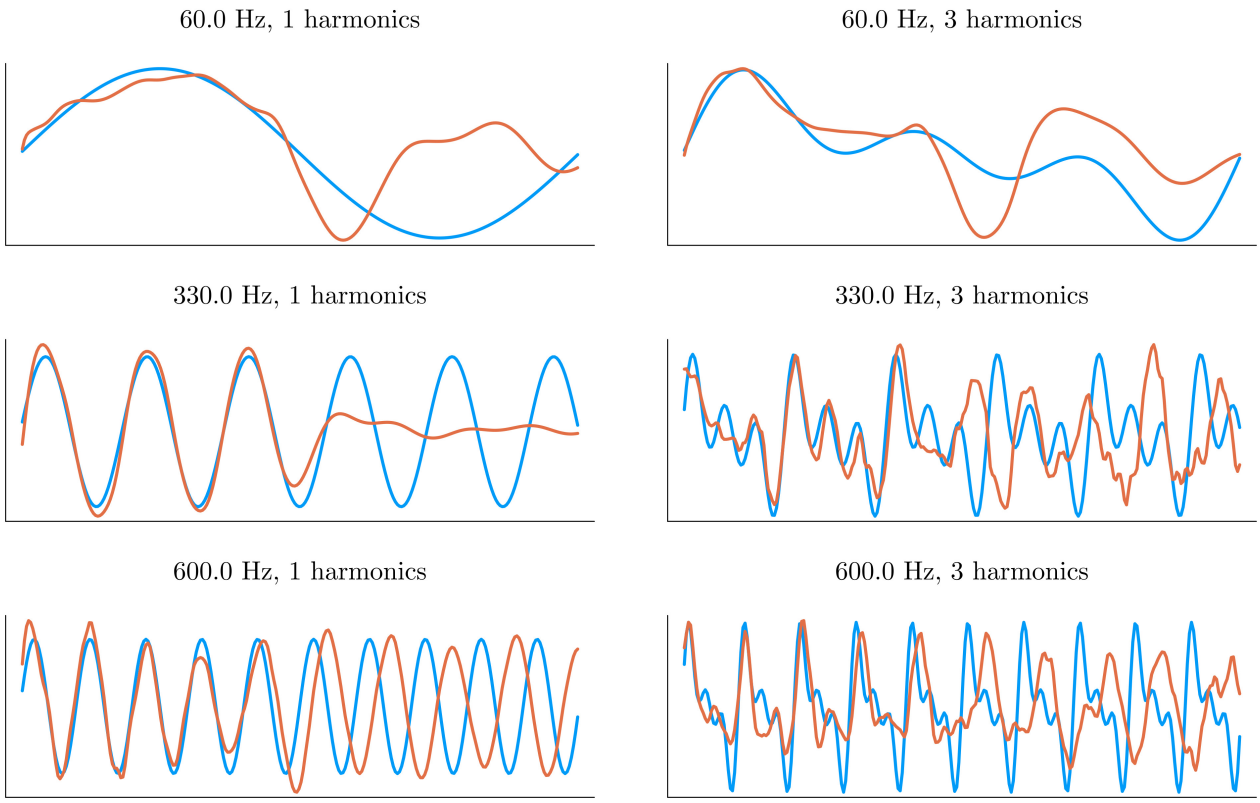


Рис. 2.22: hNODE з лінійною вкладеною моделлю NODE апроксимує гармонічні сигнали з різними частотами та числом гармонік.

наступний вигляд:

$$A \approx \begin{bmatrix} 0.1 & 0.2 & -0.7 & -1.0 & 0.1 & 0.7 & 0.2 & -0.5 & 0.5 & -0.6 \\ -0.2 & -0.2 & 0.5 & 0.3 & -1.0 & -0.4 & 0.5 & 0.7 & -0.1 & 0.5 \\ 0.7 & -0.5 & 0.0 & -0.3 & -0.5 & -0.3 & -1.6 & -0.9 & -0.3 & 0.5 \\ 1.2 & -0.6 & 0.6 & -0.1 & -0.2 & 0.1 & -0.8 & -1.0 & -1.2 & -0.1 \\ 0.4 & 0.5 & 0.6 & -0.3 & -0.3 & -0.4 & -0.2 & 0.0 & 1.1 & -0.7 \\ -0.4 & 0.9 & 0.1 & -0.1 & 0.1 & 0.0 & -0.7 & -1.5 & -1.1 & -0.3 \\ -0.2 & -0.9 & 1.2 & 0.8 & 0.1 & 1.2 & -0.3 & 0.5 & 0.7 & -0.4 \\ 0.7 & -0.3 & 0.4 & 1.2 & 0.3 & 1.8 & -0.7 & 0.3 & -0.2 & 0.3 \\ -0.6 & 0.0 & -0.1 & 0.9 & -1.0 & 1.0 & -0.8 & 0.4 & 0.0 & -0.3 \\ 0.7 & 0.1 & -0.4 & -0.1 & 0.4 & -0.5 & 0.0 & -0.3 & -0.0 & -0.2 \end{bmatrix}. \quad (2.60)$$

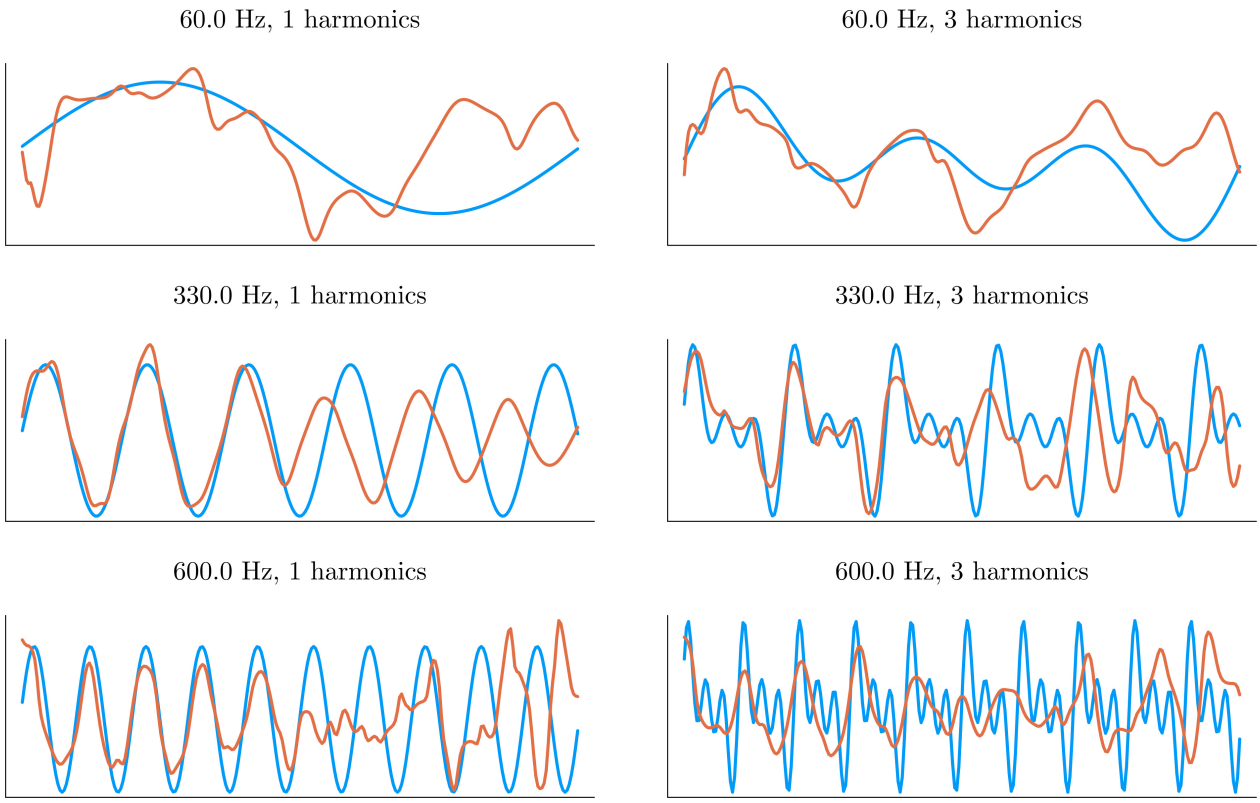


Рис. 2.23: hNODE зі вкладеною моделлю NODE, яка використовує степеневу функцію активації, апроксимує гармонічні сигнали з різними частотами та числом гармонік.

$$W \approx \begin{bmatrix} -0.7 & -0.5 & -0.6 & 0.6 & -0.0 & 1.3 & -1.3 & -1.2 & -0.5 & 0.9 \\ 0.7 & -1.0 & 0.0 & 0.2 & -0.2 & -0.5 & 0.1 & -0.9 & 0.9 & -0.8 \\ -1.1 & -0.7 & -1.0 & 0.1 & 0.1 & -0.3 & 0.7 & 0.1 & -0.2 & 1.3 \\ -0.1 & 0.5 & -0.2 & -1.3 & -0.1 & -2.3 & 2.1 & 0.9 & 1.5 & -0.4 \\ -0.7 & -0.1 & 0.4 & 0.1 & 0.4 & 0.8 & 0.7 & 0.1 & 0.8 & 0.9 \\ -0.8 & 0.9 & 0.6 & 2.2 & 0.5 & -1.6 & 0.9 & 1.3 & -0.5 & 0.2 \\ -0.8 & -0.2 & 0.3 & -0.4 & 1.3 & 0.8 & -0.7 & 1.1 & 0.8 & 2.0 \\ 0.1 & -0.8 & 0.3 & 0.0 & -1.3 & 0.6 & 0.5 & 0.2 & -1.0 & -0.8 \\ -0.4 & -1.0 & -0.5 & 0.6 & 0.4 & 1.5 & -0.4 & 0.3 & -0.5 & 0.9 \\ 0.3 & 0.9 & -0.1 & 1.1 & -0.3 & -0.5 & -0.6 & 0.1 & -0.1 & -1.1 \end{bmatrix}. \tag{2.61}$$

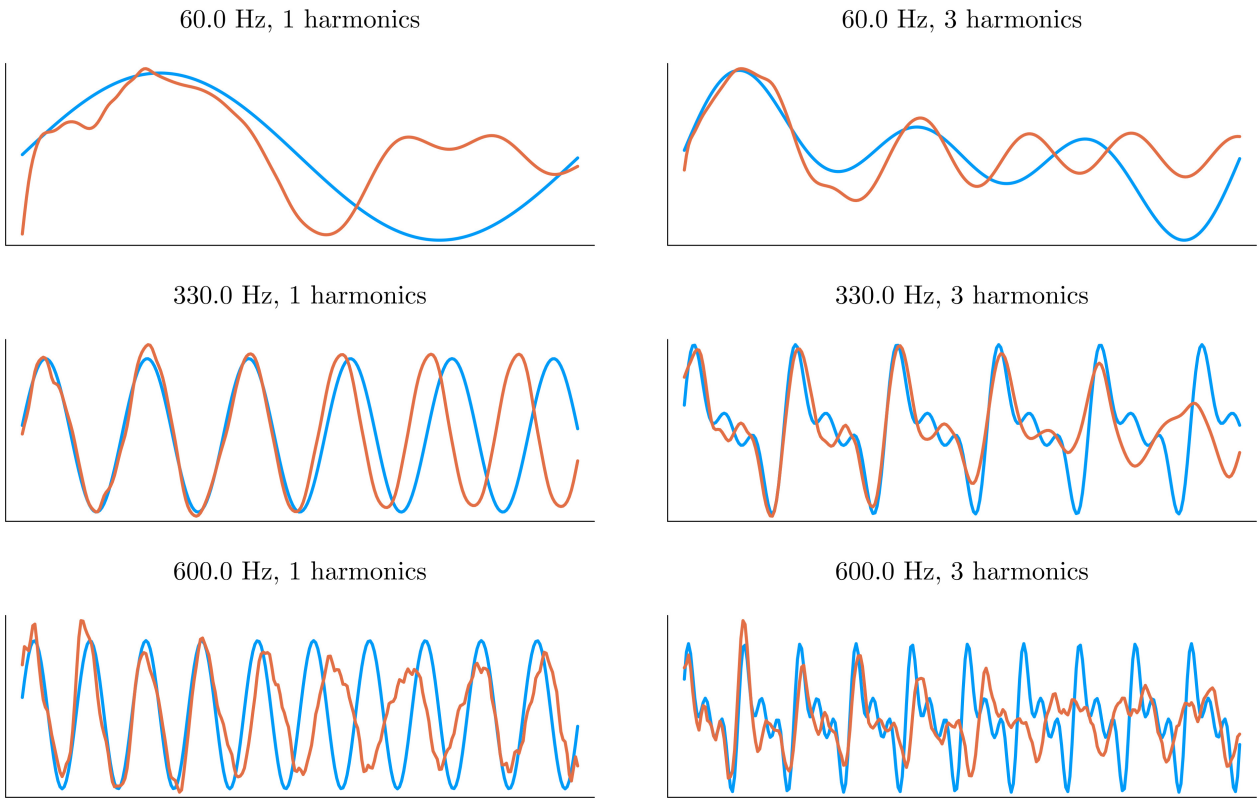


Рис. 2.24: hNODE зі вкладеною моделлю NODE, що комбінує лінійну та нелінійну функції активації, апроксимує гармонічні сигнали з різними частотами та числом гармонік.

А вектори $\mathbf{c}$ і $\mathbf{b}$ мають вигляд:

$$\mathbf{c} \approx \begin{bmatrix} 0.1 \\ 0.12 \\ -0.17 \\ 0.04 \\ 0.2 \\ -0.03 \\ 0.04 \\ -0.05 \\ 0.02 \\ 0.07 \end{bmatrix}, \quad \mathbf{b} \approx \begin{bmatrix} -0.27 \\ 0.1 \\ 0.08 \\ 0.19 \\ 0.02 \\ 0.18 \\ -0.58 \\ 0.04 \\ -0.18 \\ 0.11 \end{bmatrix}. \tag{2.62}$$

Власні числа $\Lambda = \{\lambda_1, \lambda_2, \ldots, \lambda_{10}\}$ матриць A і W наступні:

$$\Lambda_A = \begin{bmatrix} 0.14034691 \pm 1.9491125i \\ 0.001702832 \pm 0.39473188i \\ -0.05311531 \pm 3.335589i \\ -0.08532482 \pm 1.2703413i \\ -0.43842098 \pm 1.2228682i \end{bmatrix}, \; \Lambda_W = \begin{bmatrix} 1.3963987 \pm 0.90323114i \\ -0.34276614 \pm 0.56283116i \\ -0.35197243 + 0.0i \\ -1.2598333 + 0.0i \\ -1.7662214 \pm 0.9953965i \\ -2.155218 \pm 3.155014i \end{bmatrix}. \tag{2.63}$$

Значення параметрів активаційної функції:

$$\alpha = -0.5448293, \; \beta = -0.5148736, \; c = 1.0137161. \tag{2.64}$$

Графік активаційної функції з параметрами (2.64) та її похідної наведено на Рис. 2.25

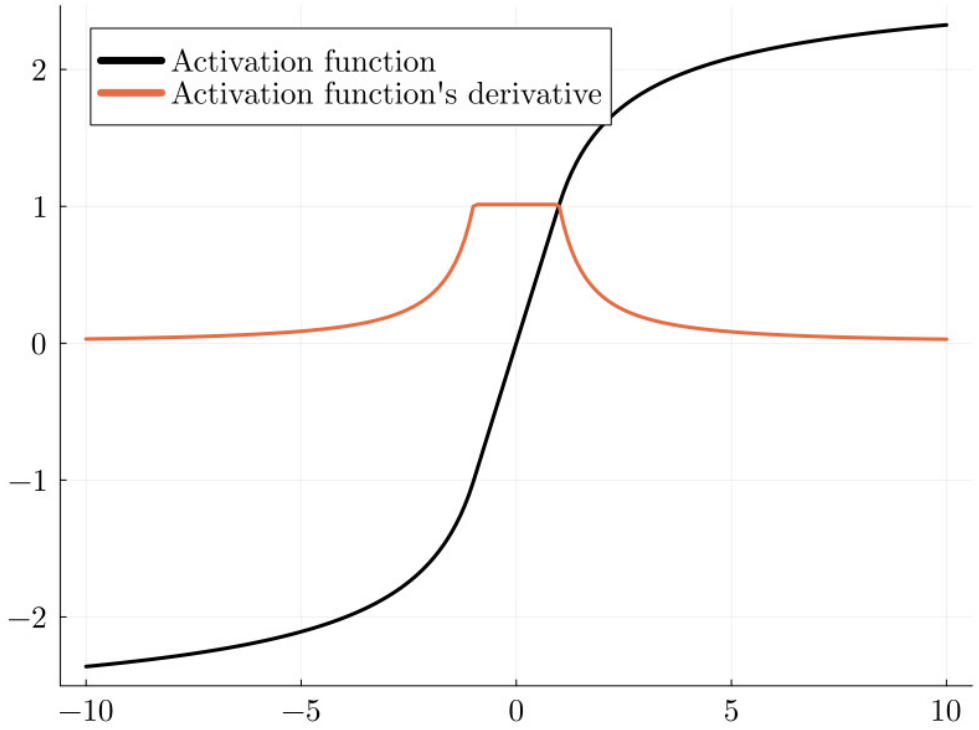


Рис. 2.25: Графік активаційної функції (2.21) з параметрами (2.64) та її похідної.

Зауважимо, що в прикладі вище параметри α і β мають негативні значення. Однак активаційна функція $w(u, \alpha, \beta, c)$ з параметрами (2.64) зберегла свої головні властивості, які були обґрунтовані для позитивних значень α і β, що значно розширює область застосування функції цього типу при розробці ней-

ромережевих моделей:

$$\lim_{w\to\infty,\alpha<0} f(w,\alpha,\beta,c) = \frac{\alpha-1}{\alpha}c^{\frac{\alpha}{\alpha-1}} < \infty,$$

$$\lim_{w\to-\infty,\beta<0} f(w,\alpha,\beta,c) = -\frac{\beta-1}{\beta}c^{\frac{\beta}{\beta-1}} > -\infty.$$

Проте за $\alpha < 0$ і $\beta < 0$ функція $w(u,\alpha,\beta,c)$ є обмеженою.

Нарешті, спектр показників Ляпунова для системи (2.28) з параметрами, описаними вище, має вигляд:

$$\Lambda = \begin{bmatrix} 0.0062366828039583325 \\ -0.07399496578897219 \\ -0.07394170378059134 \\ -0.108283715386416 \\ -0.27739753287906715 \\ -0.5804613519573166 \\ -0.5806108076254256 \\ -0.6410685713759393 \\ -2.3737137345976995 \\ -2.3741072178693563 \end{bmatrix}. \tag{2.65}$$

Спектр було обчислено за допомогою алгоритму, реалізованого в пакеті ChaosTools.jl [85] для мови програмування Julia. Алгоритм полягає у слідуванні за деякою "базовою" траєкторією системи й застосуванні QR-декомпозиції до паралелепіпеда, визначеного векторами відхилення в кожній точці траєкторії. Початкова точка для застосування алгоритму була обрана моделлю-кодувальником, і траєкторія, що їх відповідає, дала в результаті спектр показників Ляпунова, що вказує на граничний цикл. Якщо початкову точку обрати випадково, то можна отримати траєкторію, що не потрапляє в зону притягання атрактора, або навіть є нестабільною. Точка, що була обрана кодувальником із

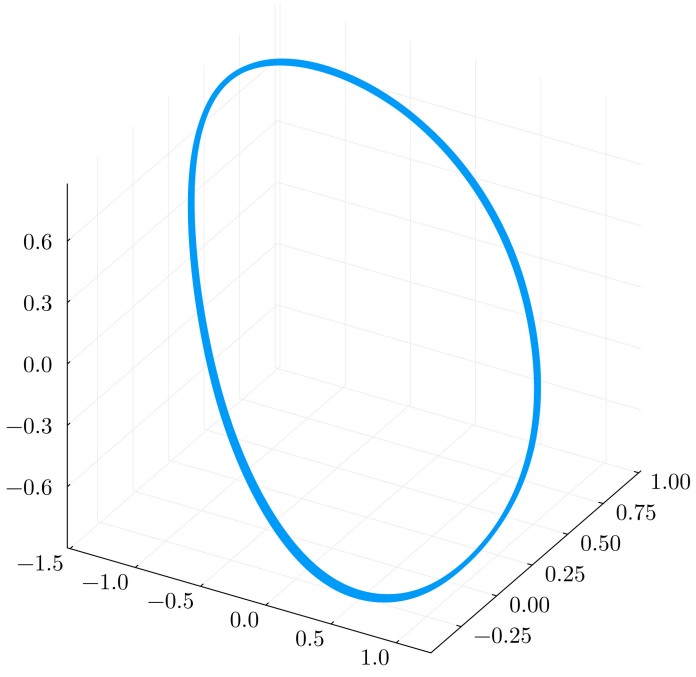


Рис. 2.26: Граничний цикл, що виникає у системі (2.28)) з параметрами (2.60), (2.61) і (2.62), а також початковими умовами (2.66).

сегмента сигналу зі 266 семплами, має наступний вигляд:

$$\mathbf{x}_0 = \begin{bmatrix} -2.9103615 \\ -2.136637 \\ -6.008354 \\ 0.690889 \\ -0.9069586 \\ 1.3819717 \\ -1.1111665 \\ -4.91214 \\ -0.14267516 \\ -0.52204686 \end{bmatrix} \tag{2.66}$$

Це є демонстрацією синергії, яка існує між компонентами hNODE моделі, що дозволяє їм покривати слабкості одне одного. Приклад траєкторії з бажаною поведінкою наведено на Рис. 2.26. Приклад траєкторій, що мають небажану поведінку, наведено на Рис. 2.27.

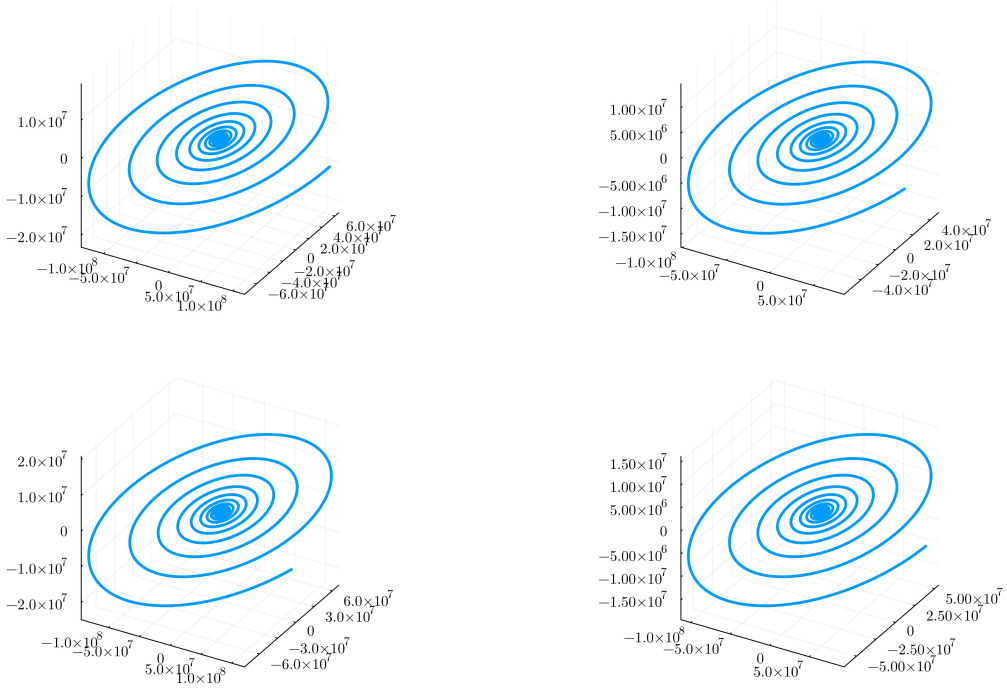


Рис. 2.27: Нестабільні траєкторії системи (2.28) з параметрами (2.60)-(2.62) з довільними початковими умовами, що лежать поза сферою з радіусом 5 навколо початку координат.

Подібний підхід до аналізу одного набору параметрів можна застосувати, щоб дослідити залежність властивостей вкладеної моделі NODE від характеристик вхідних даних. Наприклад, можна нанести на графік власні значення матриць або показників Ляпунова залежно від частоти модельованого сигналу. Графіки власних чисел матриць лінійної, нелінійної, та комбінованої моделей залежно від частоти наведено на Рис. 2.28-2.30. Графік показників Ляпунова для комбінованої системи залежно від частоти наведено на Рис. 2.31.

З наведених рисунків видно, що власні значення матриці, що задає лінійну компоненту NODE, рідко перевищують 0 (максимальне додатне значення – 0.16, що було досягнуто в матриці A комбінованої системи), що дозволяє системам досягати періодичних траєкторій. З іншого боку, власні числа матриці нелінійної компоненти зростали до 1.48, що дозволяло траєкторіям в латентному просторі мати складнішу поведінку. Значення показників Ляпунова вказують на присутність граничних циклів у системі для усіх вхідних даних.

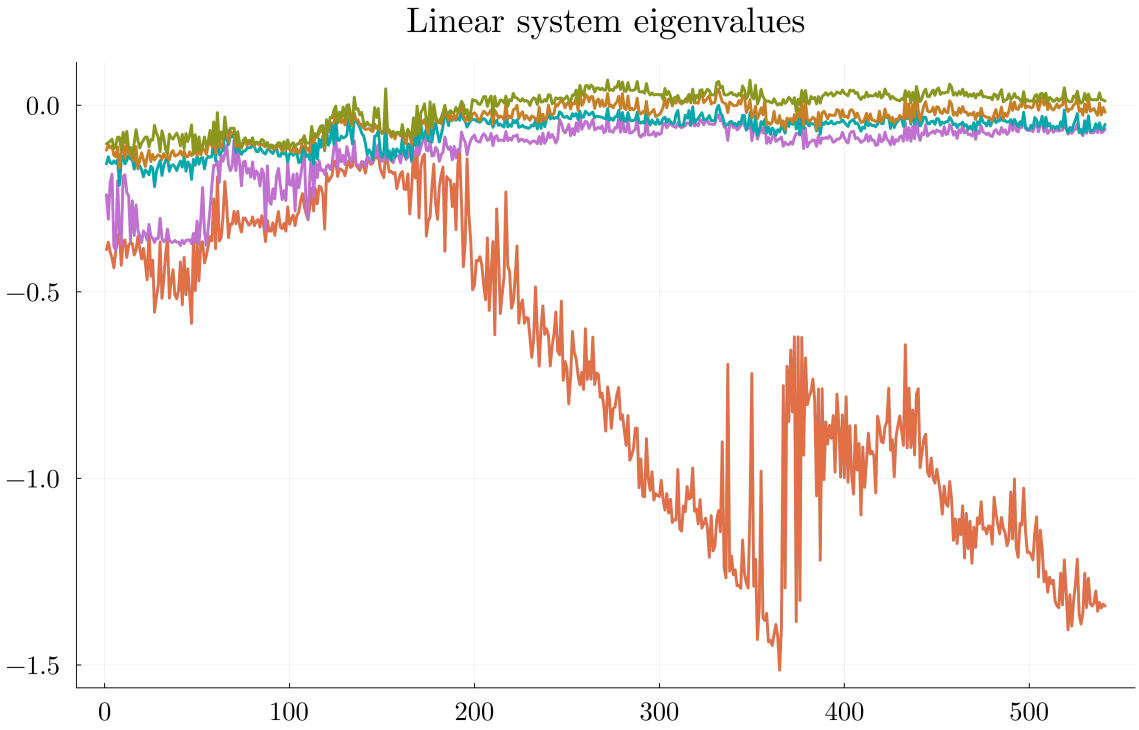


Рис. 2.28: Власні числа матриці A моделі (2.5) для вхідних даних з трьома гармоніками та частотами в діапазоні від 60 до 600 Гц.

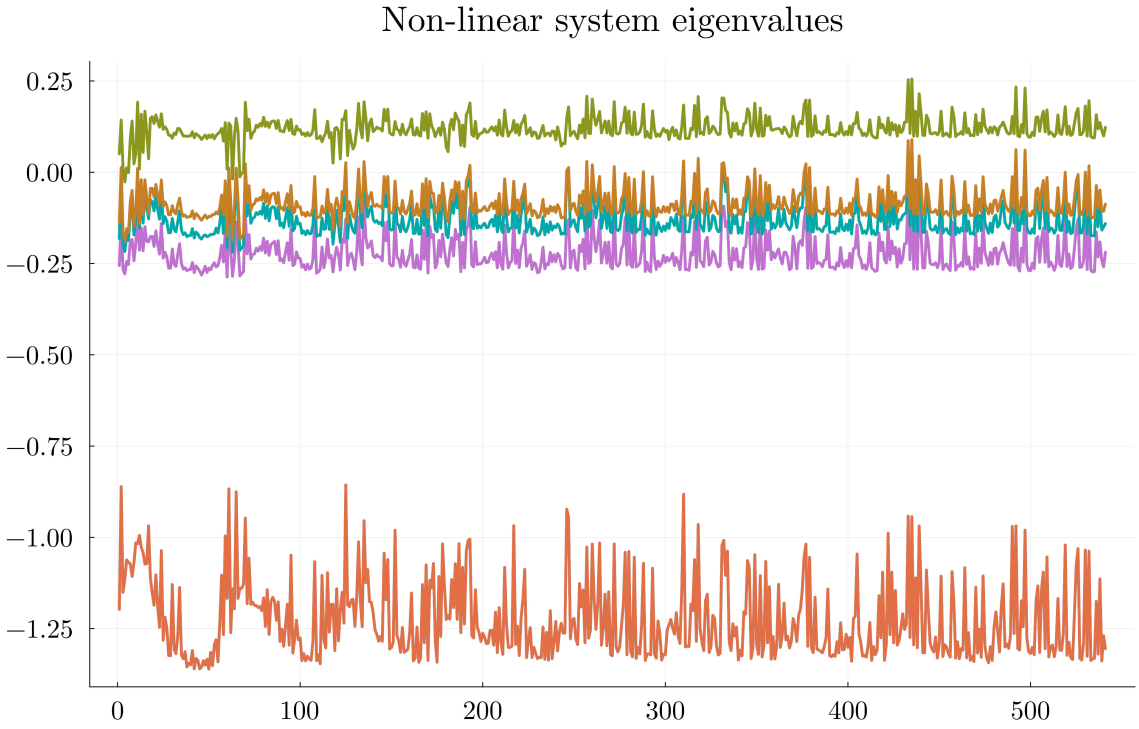


Рис. 2.29: Власні числа матриці A моделі (2.57) зі степеневою функцією у ролі функції активації для вхідних даних з трьома гармоніками та частотами в діапазоні від 60 до 600 Гц.

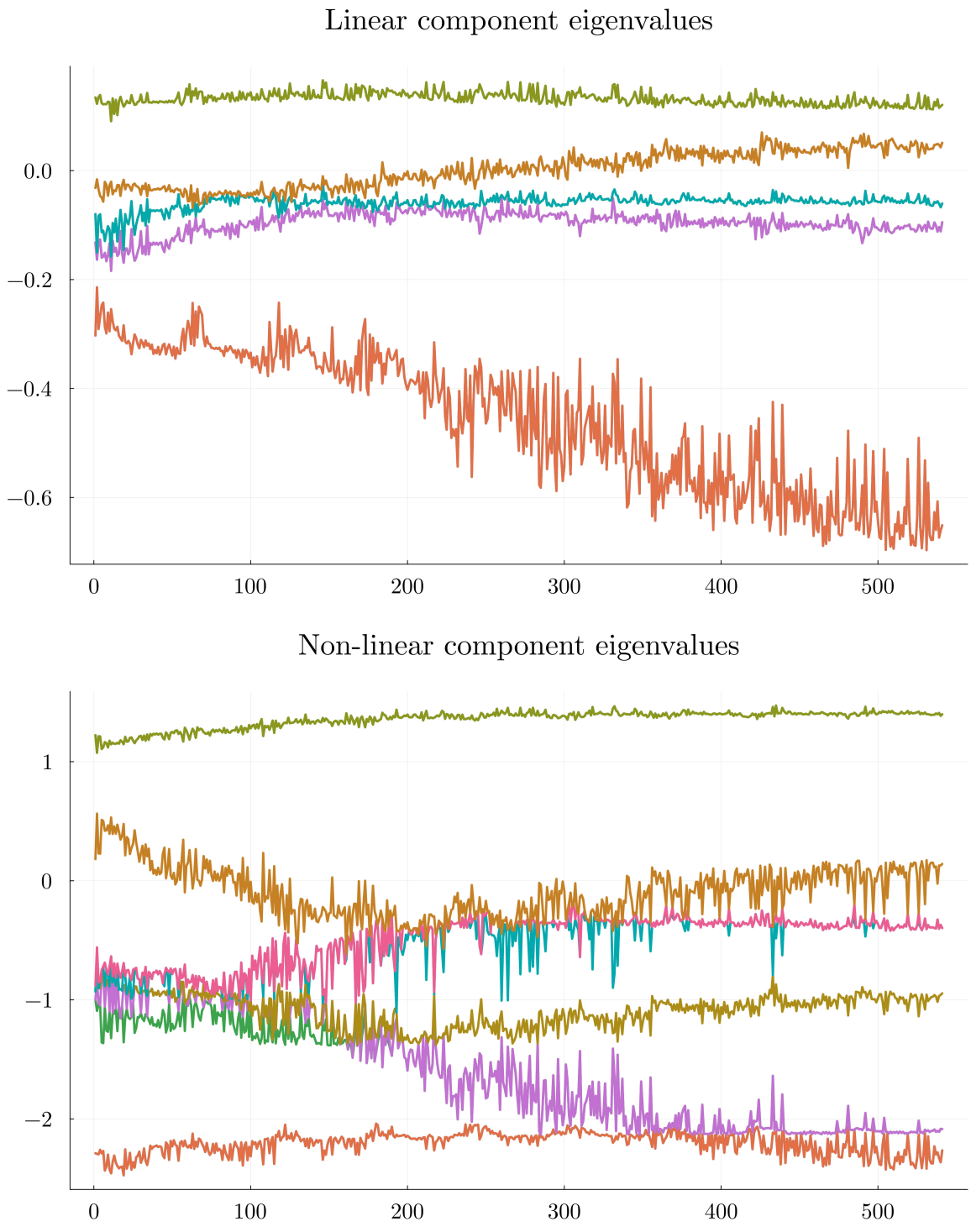


Рис. 2.30: Власні числа матриць A і W моделі (2.28) для вхідних даних з трьома гармоніками та частотами в діапазоні від 60 до 600 Гц.

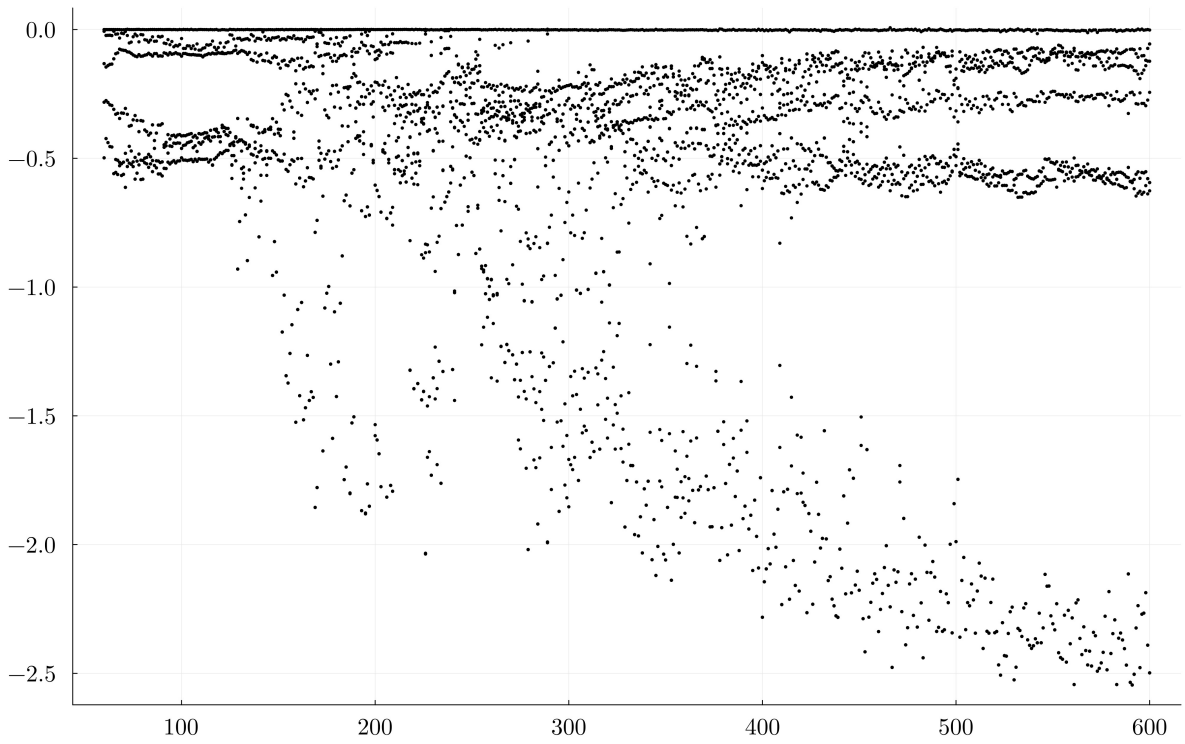


Рис. 2.31: Показники Ляпунова для моделі (2.28) для вхідних даних з трьома гармоніками і частотами в діапазоні від 60 до 600 Гц, що обчислені з початкових умов, отриманих моделлю-кодувальником.

2.7 Висновки

У розділі пропонується нова модель для апроксимації та прогнозування одновимірних нестаціонарних часових рядів за допомогою побудови адаптивного контролю для системи нейронних диференціальних рівнянь, що породжує траєкторії у багатовимірному латентному просторі, які в подальшому інтерпретуються в одновимірні точки модельованого ряду. Проведено порівняльний аналіз якості моделювання з використанням розробленої архітектури з використанням різних функцій активації, структур вкладеної системи ЗДР, та використаного кроку інтегрування. Наведено теоретичне обґрунтування вибору оптимальної функції активації та структури системи ЗДР, а також наведено умови існування точок рівноваги та обмеженості розв'язків такої системи.

Чисельні експерименти, проведені з використанням нової моделі у різних конфігураціях, показують її ефективність при розв'язанні задач апроксимації та прогнозуванні нестаціонарних часових рядів, а отже можна вважати доцільним побудову нової моделі для параметричного синтезу мовлення на основі розробленої архітектури.

Розділ 3

Моделювання голосового сигналу як динамічного процесу

3.1 Реалізація оцінювання фундаментальної частоти для потокового аудіо сигналу

Оцінка фундаментальної частоти (F_0) або тону – це процес, метою якого є визначення мінімальної частоти з якою досліджуваний сигнал повторює сам себе. Існує багато різних підходів до реалізації оцінки F_0. Ці підходи можна розділити на два широкі класи – такі, що засновані на аналізі спектра сигналу, та такі, що працюють безпосередньо з даними у часовій області. Спектральні методи оцінки фокусуються на виявленні піків та відстаней між ними у спектральній оболонці сигналу або його кепстрі. Часові методи натомість формують оцінку частоти на основі відстаней між піками та перетинами нуля сигналу у часовій області [86]. Обираючи метод оцінки F_0 для застосування у розв'язанні конкретної задачі, необхідно приділити увагу таким факторам як відношення точності до ефективності обраного алгоритму, очікуваний рівень шуму у сигналі, часові обмеження, необхідна довжина сегмента сигналу тощо. Було продемонстровано, що спектральні методи більш точні за часові [87], але вони також вимагають обчислення спектру або кепстру для кожного сегмента аналізованого сигналу, що значно підвищує їх обчислювальну складність. Саме через це

у даній роботі було обрано часовий алгоритм для оцінювання фундаментальної частоти DIO, що було розроблено як компонент фреймворка WORLD [88], оскільки він здатен швидко оцінювати F_0 сигналу, не виконуючи ніяких додаткових перетворень. Проте офіційна реалізація DIO, що написана на C++, має істотний недолік, який полягає у відсутності підтримки "покадрової"(frame-by-frame) обробки сигналів у реальному часі. У цьому розділі описано процес реалізації DIO-подібного алгоритму для роботи з покадровим аудіо.

За основу нової імплементації було взято покращену версію DIO, розроблену Даідо Р. та Хісамінато Ю. [89]. Новий алгоритм будо реалізовано на мові Julia з метою спрощення його використання в інтерактивних сценаріях дослідження даних. Нижче наведено детальний опис реалізації алгоритму. Для диференціювання оригінального алгоритму на мові C, його модифікованої версії від Yamaha, та нової реалізації мовою Julia, до назви додаватимемо префікси C, Y та J відповідно – нова імплементація DIO на Julia буде називатися JDIO, референсний алгоритм – YDIO, а C++ реалізація оригінального алгоритму з WORLD – CDIO.

Базовий алгоритм DIO включає себе три кроки – мінімізація множини можливих значень фундаментальної частоти з подальшим фільтруванням найбільш стабільних F_0 кандидатів для кожного сегменту сигналу та вибором такого кандидата, який найкраще зберігає гладкість F_0 контуру для поточного регіону. Перший крок полягає у попередній обробці сигналу що включає: децимацію (проріджження) сигналу для зменшення частоти семплування до рівня, достатнього лише для точного представлення найвищої частоти, яку алгоритм має бути здатним ідентифікувати; видалення складової постійного струму (DC component), тобто центрування вхідного сигналу так, щоб середнє значення семплів було рівним нулю; та каскадне застосування паралельних фільтрів низьких частот для видалення гармонік, щоб вони не затінювали фундаментальний тон та робили його коректну ідентифікацію більш ймовірною. Результатом такої попередньої обробки є набір з паралельних сигналів, що мають значно меншу кількість семплів в порівнянні із вхідним сигналом і середнє значення семплів, що прямує до нуля. Єдина різниця між ними – частота зрізу застосованого до них фільтру низьких частот. Другий крок алгоритму полягає у застосуванні детектора періоду до кожного з отриманих на першому кроці сигналів, а також оцінки надійності значень знайдених періодів. Останній крок, керуючись

набором евристик, направлених на мінімізацію відхилення нової оцінки F_0 від попередньої, обирає серед знайдених періодів найкращого кандидата. Схематичне зображення алгоритму наведене на Рис. 3.1.

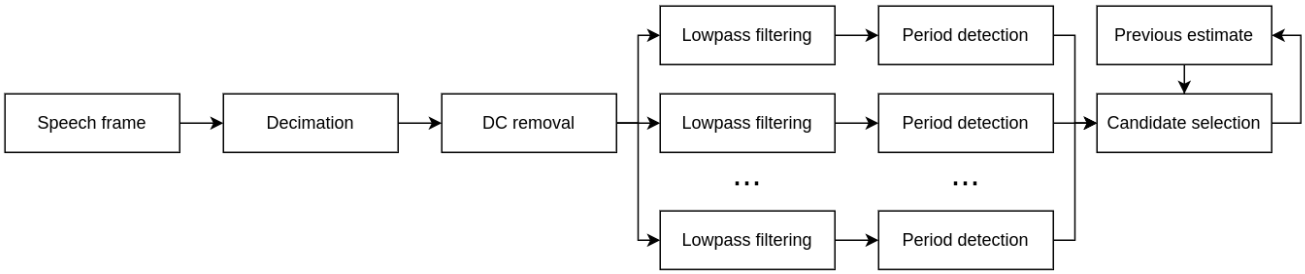


Рис. 3.1: Високорівневе представлення алгоритму DIO. Кожен його компонент може бути змінено для досягнення оптимальної продуктивності в контексті конкретної задачі.

У JDIO, децимація, вилучення компоненти постійного струму, та застосування фільтрів низьких частот було делеговано функціям пакета для роботи з цифровими сигналами DSP.jl [90], а детектори періоду та вибір F_0 кандидатів було реалізовано окремо.

Як і в YDIO, детектор періоду JDIO заснований на ітеруванні по семплах сигналу та ідентифікації проміжків часу між парами подій чотирьох типів: позитивними перетинами нуля, позитивними піками, негативними перетинами нуля, та негативними піками. Для підвищення надійності ідентифікації цих подій, JDIO шукає їх у наведеному вище порядку та за круговим принципом – наприклад, після ідентифікації негативного піка, алгоритм почне шукати позитивний перетин нуля, потім – позитивний пік і так далі. При цьому якщо подія була знайдена у час t_i, наступний пошук почнеться з часу $t_i + \tau_{min}$. Пошук вважається невдалим і припиняється, якщо до $t_i + \tau_{max}$ не було знайдено жодної події, де τ_{min} і τ_{max} – це константи, які задають жорсткі межі на максимальне і мінімальне значення можливих фундаментальних частот, які може ідентифікувати алгоритм. Пошук також припиняється у разі, коли було успішно знайдено 8 послідовних подій, тобто коли було ідентифіковано по дві події кожного типу, і отримано послідовність часових міток $[t_1, t_2, \ldots, t_8]$. З цих міток потім обчислюються чотири періоди: $p_i = t_{i+4} - t_i$, $i = \overline{1,4}$. Процес ідентифікації періоду візуалізовано на Рис. 3.3. Оцінка фундаментального періоду для поточного фрагмента сигналу оцінюється як середнє значення періодів між знайденими

подіями:

$$P_0 = \frac{1}{4}\sum_{i=1}^{4} p_i \implies F_0 = \frac{1}{P_0}. \tag{3.1}$$

Для кожної оцінки (3.1) обчислюється значення "впевненості", яке характеризує відхилення кожного з чотирьох періодів від їх середнього значення:

$$\lambda = 1 - min\left[1, \frac{\sum_{i=1}^{4} |p_i - P_0|}{4P_0}\right]. \tag{3.2}$$

Оцінки з упевненістю, що нижча за деякий поріг λ_{min}, відкидаються, а найкращий F_0 кандидат обирається з тих оцінок, що залишилися.

Базовий алгоритм доволі прямолінійний, але у реальних застосуваннях він стикається з декількома додатковими проблемами.

По-перше, ідентифікатор піків може давати високе число хибно позитивних результатів, особливо якщо сигнал, до якого застосовується алгоритм, містить шум або має велику кількість гармонік. Щоб покращити точність та уникнути таких хиб, було введено два додаткових пороги – поріг шуму $0 < \eta < 1$ та відносна амплітуда піків $0 < \alpha < 1$. Точка s_j, яка розглядається як кандидат у піки, пропускається, якщо її амплітуда менша або за шумовий поріг η або за α долю попереднього піка з протилежним знаком $s_{prev.peak}$:

$$|s_j| > \eta \ \ \& \ \ |s_j| > \alpha |s_{prev.peak}|. \tag{3.3}$$

Застосування шумового порогу також автоматично вилучає неозвучені регіони з подальшої роботи алгоритму. В іншому разі їх ідентифікація та вилучення мало б виконуватися окремим етапом роботи алгоритму.

По-друге, сигнал, для якого розроблено алгоритм, є дискретним і моменти в часі, у яких була виміряна його амплітуда, взагалі кажучи, не співпадають з істинними піками та перетинами нуля. Ця проблема стає ще гострішою після децимації сигналу, коли частота семплування стає суттєво меншою. Візуалізацію описаної проблеми наведено на Рис. 3.2. Щоб дещо пом'якшити її, було використано лінійну та квадратичну інтерполяцію. Оскільки перетини нуля шукаються як дві сусідні точки (t_j, s_j) і (t_{j+1}, s_{j+1}), які знаходяться по різні боки осі t, істинний перетин нуля може бути оцінено шляхом побудови лінійної інтерполяції між цими двома точками та знаходженні значення t, за якого

інтерполяційна пряма рівна нулю:

$$t_{zero} = -s_j \frac{t_{j+1} - t_j}{s_{j+1} - s_j} + t_j. \tag{3.4}$$

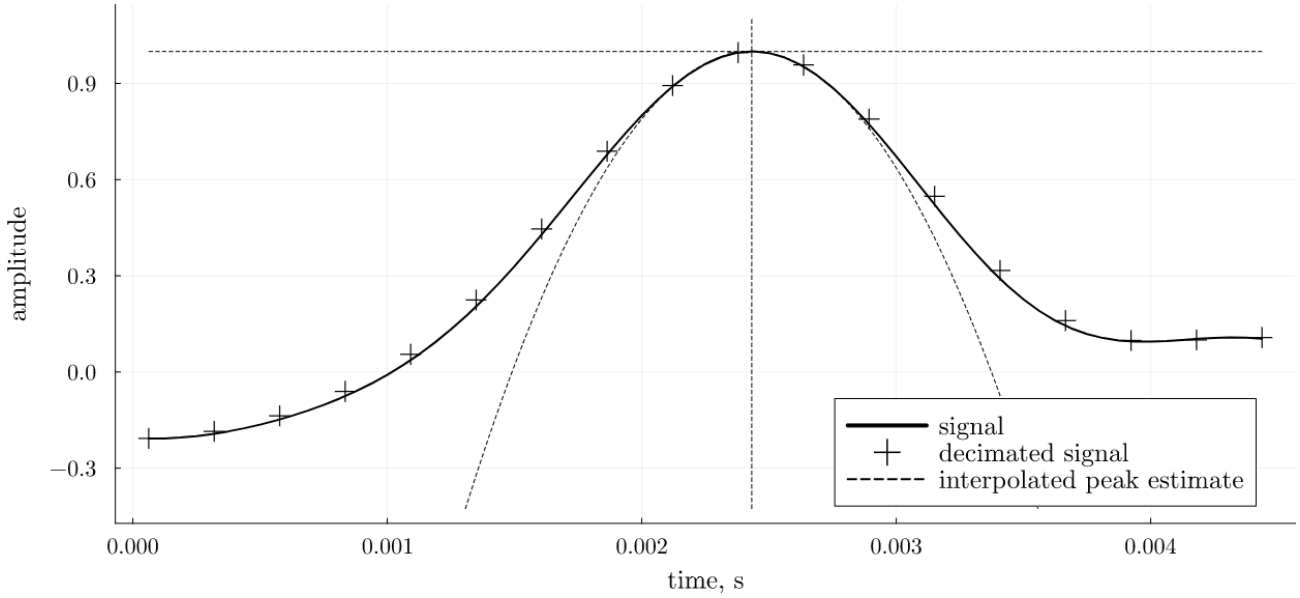


Рис. 3.2: Високорівневе представлення алгоритму DIO. Кожен його компонент може бути змінено для досягнення оптимальної продуктивності в контексті конкретної задачі.

Для виявлення піків, мають бути знайдені такі три точки (t_{j-1}, s_{j-1}), (t_j, s_j) і (t_{j+1}, s_{j+1}), що задовольняють умові $s_{j-1} < s_j \;\; \& \;\; s_{j+1} < s_j$. Істинний пік має бути більшим або дорівнювати серединній точці (t_j, s_j), і щоб знайти його апроксимацію, була використана квадратична інтерполяція:

$$\begin{bmatrix} t_{j-1}^2 & t_{j-1} & 1 \\ t_j^2 & t_j & 1 \\ t_{j+1}^2 & t_{j+1} & 1 \end{bmatrix} \begin{bmatrix} a \\ b \\ c \end{bmatrix} = \begin{bmatrix} s_{j-1} \\ s_j \\ s_{j+1} \end{bmatrix}. \tag{3.5}$$

Тут a, b, і c – коефіцієнти рівняння $f(t) = at^2 + bt + c$, яке описує параболу, побудовану на трьох точках в околиці ймовірного піка. З цього рівняння маємо координати екстремума, які й обираємо в якості апроксимації істинного піка:

$$\begin{aligned} t_{peak} &= -\frac{b}{2a} \\ s_{peak} &= -\frac{b^2}{4a} + c. \end{aligned} \tag{3.6}$$

Важливо знайти не лише часову мітку t_{peak}, але і значення амплітуди s_{peak},

щоб його можна було використати при перевірці умови (3.3) для наступного піка.

По-третє, оскільки декілька детекторів періоду застосовуються до декількох паралельних сигналів, отриманих шляхом застосування різних фільтрів низьких частот до вхідного сигналу, результатом роботи алгоритму можуть бути декілька F_0 кандидатів $C = \{c_1, c_2, \ldots, c_k\}$ та асоційованих з ними значень впевненості $\Lambda = \{\lambda_1, \lambda_2, \ldots, \lambda_k\}$, серед яких необхідно обрати найбільш правдоподібне значення. JDIO використовує підхід, який максимізує значення впевненості λ нового кандидата та мінімізує різницю між попередньою та поточною оцінками F_0:

$$F_0^n = \begin{cases} c_i, & \text{if } \lambda_i = \max(\Lambda) \ \ \& \ \ \lambda_i \geq \lambda^{n-1} \\ \underset{c \in C}{\operatorname{argmin}}(c - F_0^{n-1}), & \text{otherwise.} \end{cases} \tag{3.7}$$

Значення впевненості λ гарантовано задовільні, оскільки будь-які значення нижче заданого порогу λ_{min} відкидаються. Якщо не було знайдено такого кандидата, асоційована впевненість якого більша або дорівнює впевненості у минулому кандидаті, обирається значення F_0, яке найменш відрізняється від попереднього. Такий підхід допомагає запобігти помилкам, коли значення шуканої частоти подвоюється або зменшується вдвічі при аналізі гармонічних сигналів.

По-четверте, оскільки алгоритм DIO зазвичай використовується для оцінки контуру фундаментальних частот для сегментів з дуже коротким часом стрибка, необхідно, щоб ланцюг виявлених подій розташовувався якомога ближче до початку поточного сегмента. Щоб досягти цього, JDIO застосовує кожен тип детектора на самому початку сегмента, і обирає у якості початкового той, який виявив свою подію раніше за інших. Приклад того, як різні детектори обираються першими у ланцюжку виявлення, наведено на Рис. 3.3. На Рис. 3.3(a) першою виявленою подією є позитивний перетин нуля, а на Рис. 3.3(b) видно, що першою подією став позитивний пік, який задовольняє умові (3.3). В обох випадках алгоритму вдалося обрати найкращий ланцюг подій, що був найближче до початку сегмента і забезпечив умови для коректної оцінки фундаментальної частоти.

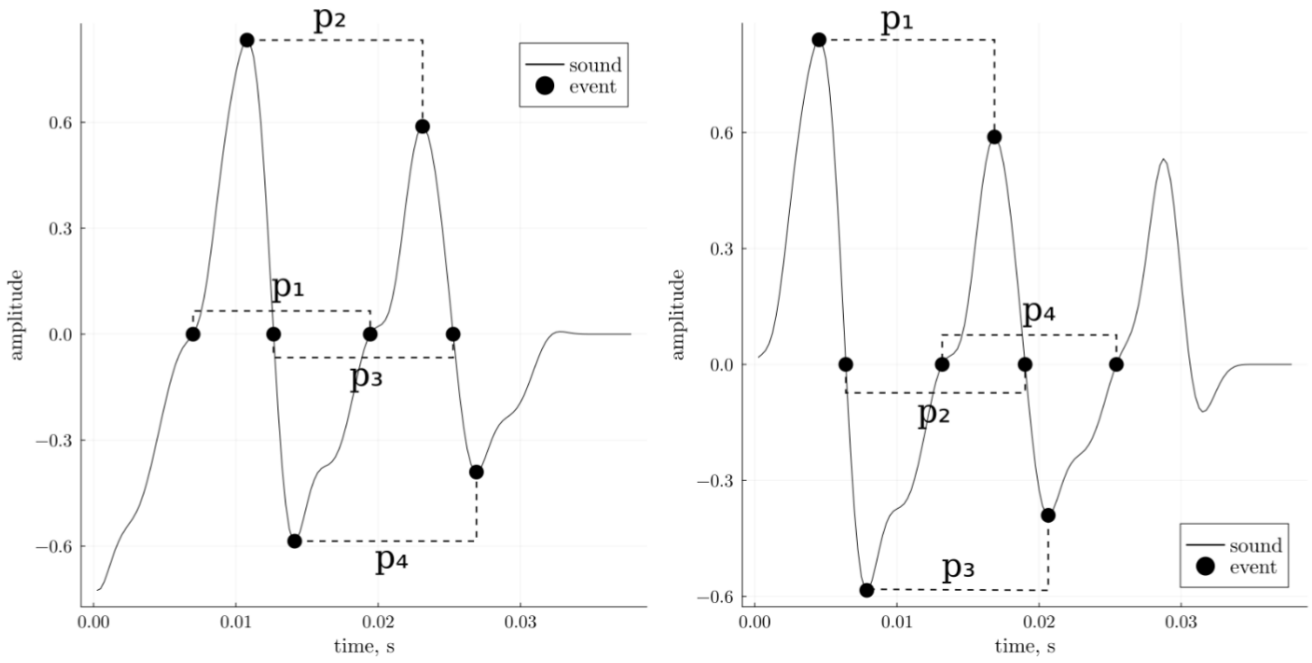


Рис. 3.3: JDIO виявляє періоди між різними типами подій у сегменті сигнала.

3.2 Реалізація оцінювання спектральних характеристик голосу для потокового аудіо сигналу

Спектральна оболонка – це крива на графіку залежності амплітуди від частоти, що отримується шляхом згладжування частотного спектра перетворення Фур'є для вхідного сигналу. В ідеальному випадку, спектральна оболонка – це гладка лінія з мінімумом коливань, що тісно обволікає спектр, поєднуючи його піки одне з одним [91]. У більшості практичних задач з обробки сигналів, спектр отримується із короткого сегмента вхідного сигналу, до якого було застосовано віконну функцію, та який було та розширено нулями для отримання бажаного розширення у просторі частот. Таке перетворення називається віконним перетворенням Фур'є (short-time Fourier transform, STFT). Послідовне застосування цього перетворення на суміжних сегментах сигналу, що перетинається, утворює спектрограму даного сигналу. Спектрограма описує зміну якості сигналу з плином часу, але через високе розширення у просторі частот, вона може містити надлишкову інформацію та шум. Оцінка спектральної оболонки для кожного STFT допомагає згладити спектр та надати більш ясну картину великих стру-

ктур та їх змін з часом. Спектральну оболонку також легше представити в параметричному вигляді, що дозволяє ефективно використовувати її в методах стиснення сигналів та надійного виявлення визначних ознак [92].

Фреймворк WORLD використовує алгоритм CheapTrick для оцінки спектральної оболонки для кожного сегменту вхідного сигналу. Версія цього алгоритму, що реалізована на C++ як частина WORLD, в подальшому тексті буде називатися CCheapTrick. При роботі зі статичними звуковими файлами CCheapTrick дійсно видає стабільні та якісні результати, але при його послідовному застосуванні на потоці сегментів сигналу в реальному часі ця стабільність за невідомих причин втрачається і призводить до суттєвих викривлень та шуму у реконструйованому сигналі. Порівняння результатів роботи CCheapTrick на потоковому аудіо та на передзаписаному звуковому файлі наведено на Рис. 3.4.

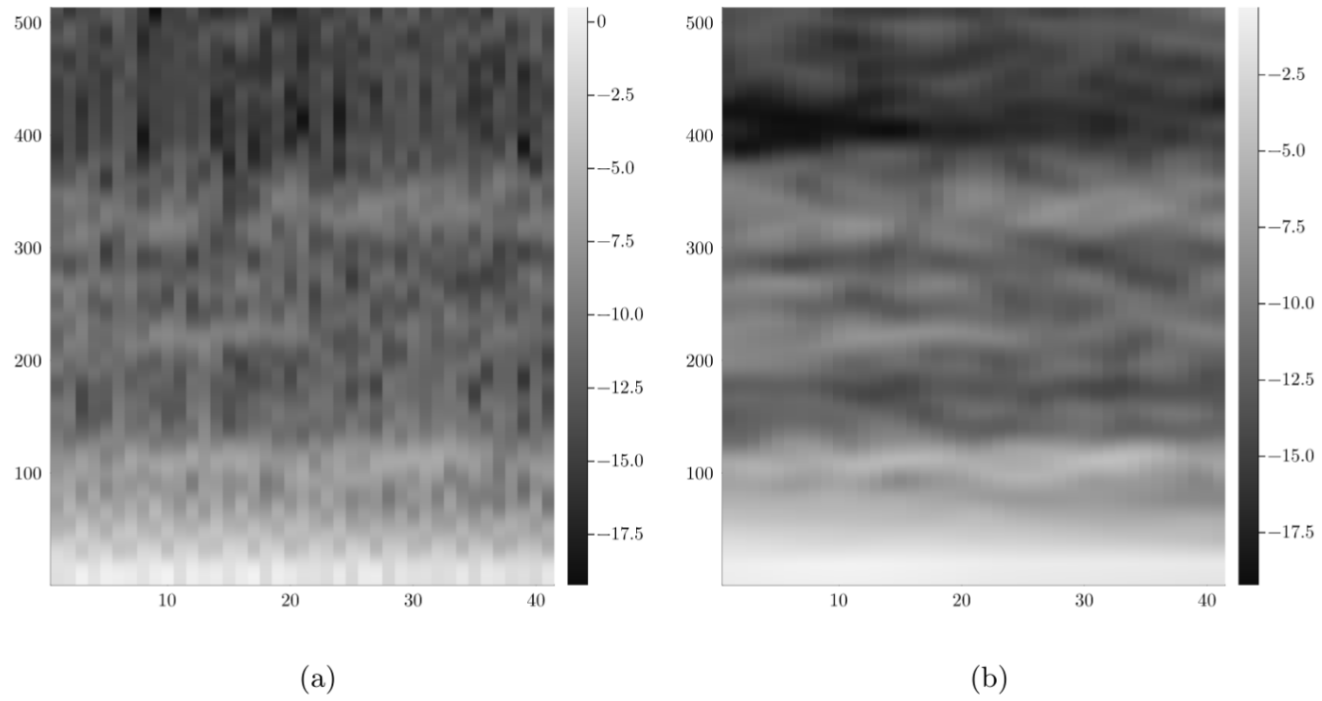


Рис. 3.4: Використання CCheapTrick для оцінки спектральної оболонки для сегментів потокового аудіо (a) та для передзаписаного файлу (b).

Нова реалізація цього алгоритму – JCheapTrick – була написана на мові Julia, щоб дозволити його інтерактивне застосування для аналізу та синтезу сигналів в реальному часі у поєднанні з описаною вище реалізацією JDIO. Нижче наведено детальний опис алгоритму, а також мотивацію прийнятих під час дослідженні та імплементації рішень.

Алгоритм CheapTrick заснований на адаптивному сегментуванні сигналу, яке підлаштовує розмір вікна до значень моментальної частоти поточного сегмента. У якості віконної функції застосовується вікно Ханна [93], що задається квадратом косинуса на відрізку $[-0.5, 0.5]$. Розмір вікна обирається таким чином, щоб воно могло вмістити в себе три періоди поточної фундаментальної частоти за відповідної частоти семплування – $w_n = 3f_s F_0^{-1}$. Наприклад, якщо поточний сегмент містить 800 семплів сигналу, семплованого з частотою 16 кГц, а поточна частота становить 220 Гц, то CheapTrick обере довжину вікна 218, тобто лише 218 семплів з початку поточного сегмента буде використано для обчислення спектра. Такий підхід забезпечує фундаментальній частоті та її гармонікам домінантну роль у спектрі, оскільки частота визначається якомога ближче до початку сегмента, як показано на Рис. 3.3. До зменшеного сегмента потім застосовується віконна функція Ханна і до результату додаються нулі, щоб забезпечити необхідне розширення перетворення Фур'є та гарантувати однаковий розмір усіх спектрів незалежно від обраного розміру вікна. Це забезпечує можливість об'єднання обчислених спектрів у спектрограму.

Після обчислення спектра для поточного сегмента, магнітуда кожної з частотних комірок масштабується шляхом ділення її на суму значень вікна Ханна для збереження сумарної енергії сегмента. Масштабований спектр потім проходить через декілька згладжувальних трансформацій.

Перша трансформація – застосування звичайного FIR фільтру з прямокутним вікном. Довжина вікна фільтра обирається як індекс частотної комірки в спектрі, до якої належить поточна оцінка фундаментальної частоти. Таким чином, спектр для сегмента з меншою локальною частотою буде згладжено з використанням меншого вікна і навпаки. Іншими словами, розмір вікна фільтра обирається так, щоб воно не було більшим за відстань між гармоніками фундаментальної частоти, виміряну в частотних комірках перетворення Фур'є. Фільтр з обраним вікном застосовується двічі в прямому та зворотному напрямках, щоб уникнути фазових спотворень у результаті, тобто щоб оригінальний спектр не зсунувся в жодну сторону після застосування фільтра.

Друга трансформація – застосування композитної функції-ліфтера до кепстра сигналу для отримання фінальної спектральної оболонки.

Кепстр – це "спектр спектра", який обчислюється як перетворення Фур'є логарифмів компонентів спектра оригінального сигналу. Так само як спектр

допомагає візуалізувати та виявляти періодичності в сигналах часової області, кепстр виявляє періодичності у самому спектрі. Наприклад, для гармонічного сигналу з фундаментальною частотою $F_0 = 220$ Гц буде мати спектр з рівновіддаленими одне від одного піками, що вказують на F_0 та її гармоніки; кепстр, отриманий з цього спектра, буде мати пік на F_0^{-1}, що вказуватиме на сильну періодичність в спектрі. Оскільки кепстр отримується звичайним перетворенням Фур'є, усі техніки, які можуть бути застосовані до спектра, актуальні й для кепстра. Наприклад, на кепстрі можна застосовувати фільтри низьких та високих частот, щоб змінити характеристики відповідного йому спектра. Кепстральні фільтри називають ліфтерами, а процес їх застосування – ліфтерінгом.

Алгоритм CheapTrick використовує наступні функції для ліфтерінгу:

$$l_s = \begin{cases} \dfrac{\sin(F_0\pi\tau)}{F_0\pi\tau}, & \text{if } \tau > 0 \\ 1, & \text{if } \tau = 0 \end{cases}, \tag{3.8}$$

$$l_q = q_0 + 2q_1\cos(2\pi\tau F_0). \tag{3.9}$$

Тут τ – значення кепстральної частоти, q_0 і q_1 – константи, значення яких було визначено шляхом оптимізації і які рівні 1.18 і -0.9 відповідно. Функція (3.8) виконує роль фільтра низьких частот, який вилучає зі спектра усі компоненти, які відповідають фундаментальній частоті та її гармонікам, а також сприяє придушенню темпоральних нестабільностей у спектрі, що виникають через сегментування сигналу та взаємодії сегментів зі згортками. Функція (3.9) відповідає за те, щоб перетворення дотримувалися вимог теорії послідовного семплування, а саме щоб цифровий сигнал був незмінним після його перетворення в аналоговий і назад. Оскільки обидві ці функції обчислювально дорогі, реалізація JCheapTrick обчислює лише значення фільтрів лише до індексу F_0^{-1} кепстральної частоти, а решту значень заповнює нулями. З цим змінами, об'єднана функція ліфтерінгу набуває вигляду:

$$L(\tau) = \begin{cases} l_s(\tau)l_q(\tau), & \text{if } \tau < F_0^{-1} \\ 0, & \text{if } \tau \geq F_0^{-1} \end{cases}. \tag{3.10}$$

Застосування такого ліфтера дозволяє отримати темпорально стабільний спектр.

3.3 Оцінка якості розроблених методів оцінювання характеристик мовлення

JDIO та JCheapTrick були розроблені як альтернатива CDIO та CCheapTrick для уможливлення роботи з сигналами в реальному часі без втрати якості та точності, але оскільки CDIO та CCheapTrick не здатні працювати з короткими сегментами аудіо, пряме порівняння двох реалізацій неможливе. Через це контури F_0 та спектрограми тестових аудіо файлів були отримані C++ версіями алгоритмів без розбиття, у той час, як Julia версії оброблювали файли сегмент за сегментом, симулюючи потік даних в реальному часі, а їх вивід було конкатеновано для отримання результатів в такому ж форматі, у якому їх вивели CDIO та CCheapTrick, щоб зробити пряме порівняння можливим.

Для порівняння якості оцінки контуру фундаментальної частоти було згенеровано ряд синтетичних сигналів. У таблиці 3.1 наведено порівняння відхилень оцінок значень F_0 алгоритмами CDIO та JDIO від істинних значень частоти.

Сигнал	JDIO	CDIO
Синус з постійною частотою	0.003	242.0
Синус з постійною частотою та амплітудною модуляцією	0.008	242.0
Синус з лінійною частотною модуляцією від 60 до 600 Гц	7.8	20.4
Синус з частотною та амплітудною модуляціями	7.8	20.5
Гармонічний синус з амплітудною модуляцією	67.3	20.7
Гармонічний синус з модуляціями та шумом	14716.1	25517.0

Табл. 3.1: Порівняння середньоквадратичних відхилень контурів F_0 від істинних значень частот синтетичних сигналів для JDIO та CDIO.

З наведених значень видно, що точність CDIO підвищується з підвищенням складності сигналу, але значно деградує при наявності шуму в сигналі. Водночас поведінка JDIO є протилежною. Однак тут важливі не самі значення помилок, а їх розподіл вздовж усього сигналу. Точність CDIO в усіх порівняльних тестах була стабільною, а головний внесок у високі значення помилок зробили оцінки фундаментальної частоти на початках та у кінцях контурів F_0. Наявність таких помилок пояснюється наявністю багатоетапної постобробки, яку використовує CDIO для вибору оптимальних значень F_0 зі знайдених кандидатів вздовж усього контуру – кандидати на початку та у кінці контуру не мають достатнього контексту для формування коректної оцінки, через що постобробка

не може дати оптимальної оцінки. Значення F_0, що значно віддалені від границь сигналу, оцінюються CDIO дуже точно, але ця точність не зберігається для високих частот. JDIO не використовує жодних форм постобробки, окрім вибору оптимального кандидата F_0 для поточного сегмента, ґрунтуючись на оцінці для попереднього сегмента, описаній в (3.7), та дає стійкі оцінки вздовж усього сигналу. Іншими словами, контур фундаментальних частот, обчислений JDIO, буде мати однакову форму, не залежно від того, з якого семплу в сигналі він почав працювати. Графічне зображення розподілу помилок наведено на Рис. 3.5 та 3.6.

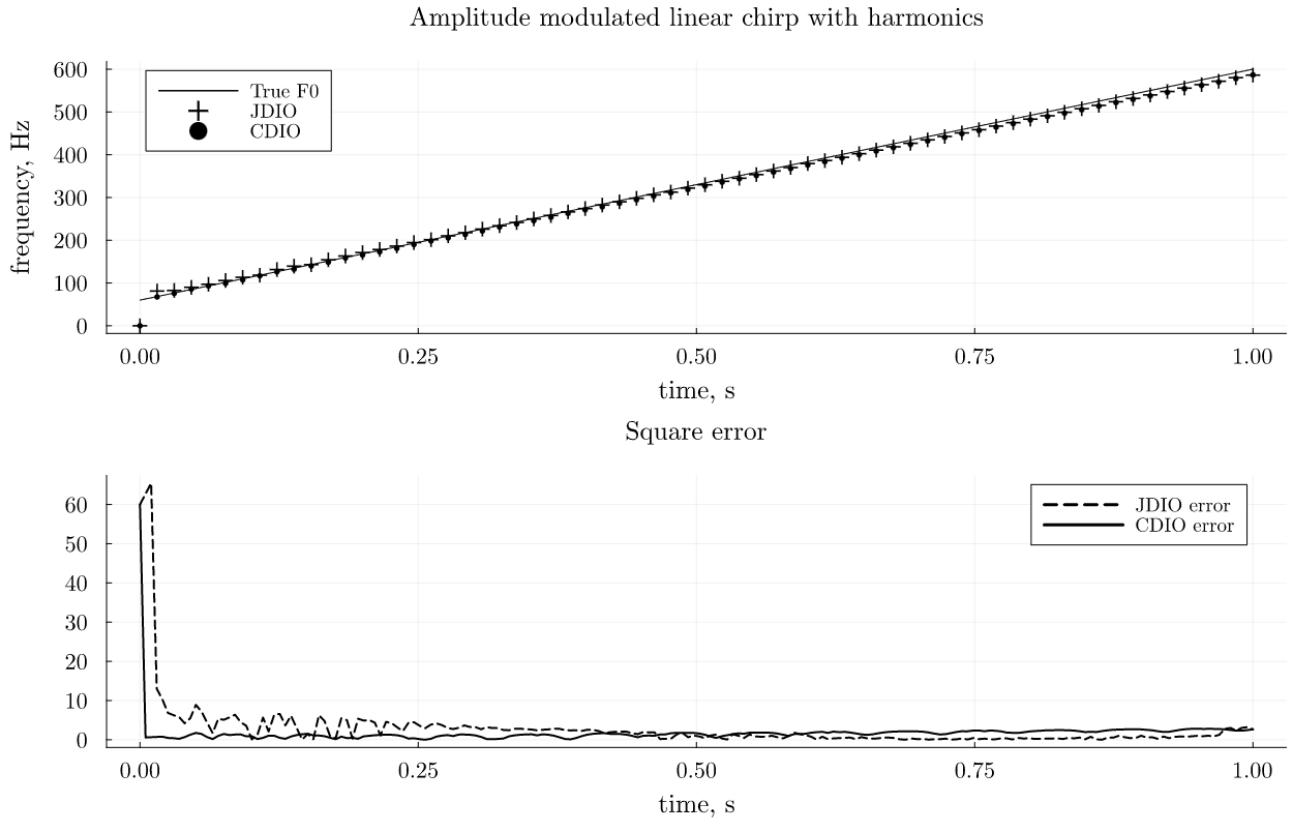


Рис. 3.5: Точність та відхилення оцінок фундаментальної частоти гармонічного сигналу з частотною модуляцією для CDIO та JDIO.

Оцінка контуру F_0 є доволі простою процедурою, оскільки істинні значення шуканих частот можна легко отримати з параметрів синтезованого сигналу. Однак для спектральних оболонок істинні значення складно визначити навіть для синтетичних сигналів. Тож для оцінки точності роботи JCheapTrick відносно CCheapTrick, використаємо непрямий підхід:

1. Оцінимо контур F_0 за допомогою JDIO та CDIO.

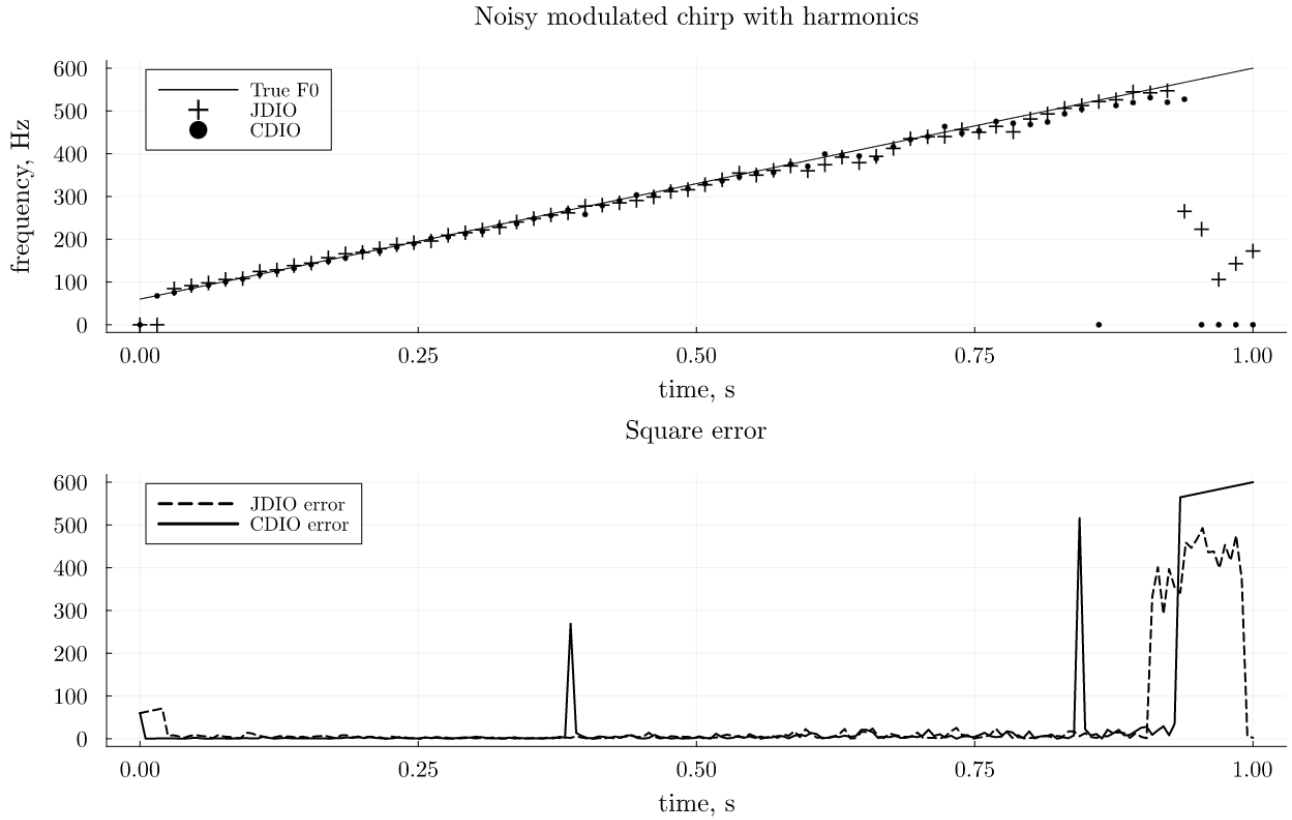


Рис. 3.6: Точність та відхилення оцінок фундаментальної частоти зашумленого гармонічного сигналу з частотною модуляцією для CDIO та JDIO.

2. Отримані контури використаємо для оцінки спектральних оболонок за допомогою JCheapTrick та CCheapTrick відповідно.

3. Обчислимо контур аперіодичності для обох отриманих спектрограм.

4. Використаємо контури F_0, спектрограми, та контури аперіодичності, щоб синтезувати аналізований сигнал за допомогою вокодера (параметричного синтезатора мовлення), реалізованого в WORLD.

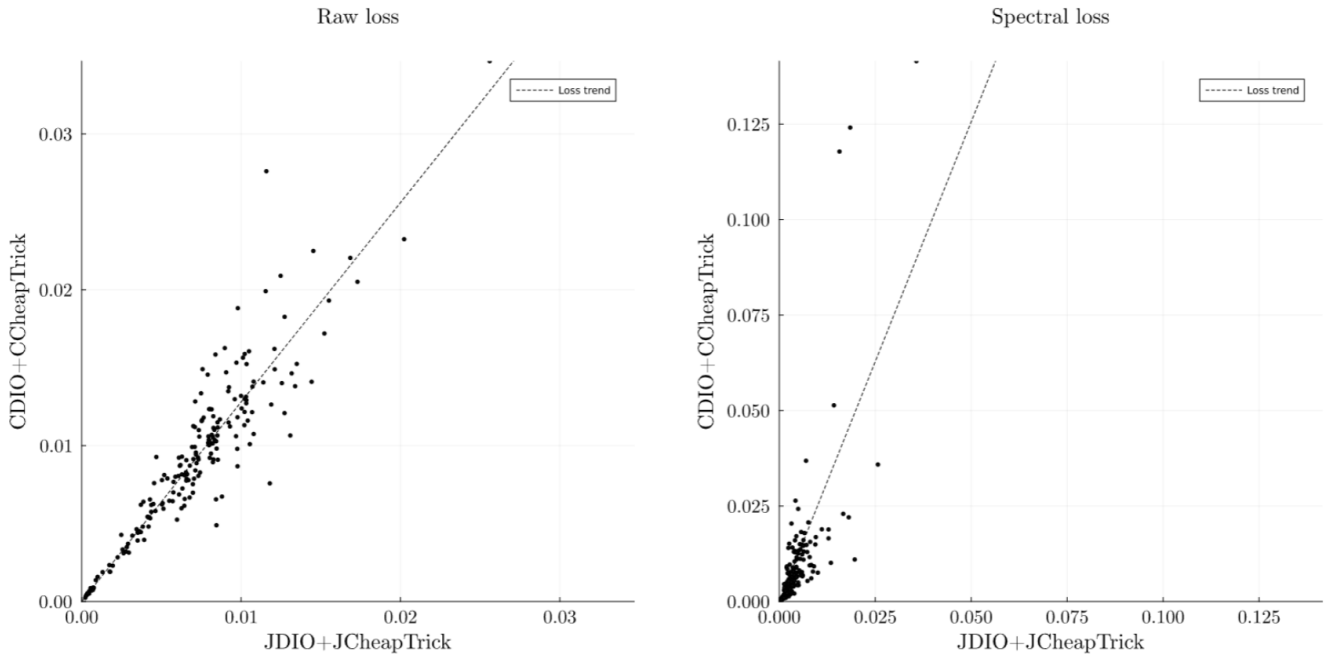
5. Знайдемо похибку як середньоквадратичне відхилення синтезованого сигналу від оригінального.

6. Знайдемо спектральну похибку як середньоквадратичне відхилення спектрограми синтезованого сигналу від спектрограми оригіналу.

Цей підхід також дозволить продемонструвати сумісність реалізацій JDIO та JCheapTrick із фреймворком WORLD.

Замість синтезованих сигналів, у цьому порівнянні використовувались реальні записи мовлення зі збірника LibriSpeech [84]. Значення часових та спе-

ктральних похибок наведені на Рис. 3.7



(a) (b)

Рис. 3.7: Значення часових та спектральних похибок ресинтезованого мовлення відносно оригінального сигналу. Значення нахилів серединних ліній становлять 1.27 для часової та 2.51 для спектральної похибки

Похибки показують, що JDIO та JCheapTrick дозволила вокодеру ресинтезувати мовлення з більшою точністю – середньоквадратичне відхилення виявилося в ≈ 1.27 менше, а спектральні характеристики ресинтезованого мовлення були точнішими в ≈ 2.51 рази у порівнянні з CDIO та CCheapTrick.

3.4 Вибір та реалізація методу переходу від одновимірного часового ряду до багатовимірного латентного простору та назад

Аналіз часових рядів – це процес добування інформації зі впорядкованої множини точок з метою отримання даних про систему, що їх породила, або передбачення майбутніх значень ряду. Найпростіша форма часового ряду – однофакторний ряд, що являє собою послідовність одновимірних точок. Якщо система, з якої були зняті такі вимірювання, достатньо складна, часовий ряд може мати періодичну, квазіперіодичну, хаотичну, та інші види поведінки. Якщо по-

будувати графік значень ряду відносно часу, усі ці види можна відносно легко ідентифіковані “на око”, але такий підхід навряд можна кваліфікувати як належний аналіз. На щастя, у багатьох галузях науки та інженерії були розроблені методи для об’єктивної ідентифікації та оцінки різноманітних властивостей часових рядів. Проте багато з цих методів мають суворі вимоги щодо розмірності задачі та можуть давати абсурдні результати, якщо розмірність конкретного ряду, до якого застосовується метод, завелика або замала. Цей тип проблем може бути вирішена за допомогою вкладення наявних даних у більш високорозмірні простори, у яких можна коректно застосувати бажані методи.

Вкладення часових рядів – це відображення сегментів даних ряду в деякий векторний простір $\mathbb{R}^n$. Це відображення має бути ін’єктивною функцією, що продукує гладку траєкторію з векторів вкладення для суміжних сегментів часового ряду. У цьому розділі наведено огляд різних методів вкладення часових рядів, розглянуто як кожен з них відповідає описаним вище вимогам, та проведено порівняльний аналіз того, наскільки добре кожен метод зберігає якісні характеристики системи, що було використано для породження часового ряду.

Широке визначення вкладення часового ряду дозволяє багато гнучкості у виборі функції відображення. Нижче наведено огляд деяких традиційних методів для переходу від часового ряду до простору вкладення.

Найбільш відомим способом побудови вкладення для даних часового ряду, що підходить під наведене вище означення, є застосування до них віконного перетворення Фур’є (STFT) або споріднених йому методів вейвлетних перетворень (wavelet transformations, WTs), які широко застосовуються для визначення ознак різноманітних сигналів [94]—[96] завдяки переходу від часової до частотної області для представлення послідовностей сегментів сигналу. STFT алгоритм має наступні кроки:

1. Сигнал розбивається на короткі сегменти (вікна), що перетинаються.

2. До кожного сегменту застосовується віконна функція, що пригнічує високочастотні спотворення, що можуть виникати на його межах.

3. Оброблений віконною функцією сегмент доповнюється нулями до такої довжини, що забезпечує бажану роздільну здатність у просторі частот та відповідає вимогам для застосування швидкого перетворення Фур’є

(fast Fourier transform, FFT) для досягнення максимальної обчислювальної ефективності.

4. На підготовленому попереднім кроком сегменті застосовується дискретне перетворення Фур'є.

5. Обчислюються квадрати абсолютних значень першої половини результату перетворення, що дає характеристику відносних інтенсивностей частот, що складають сегмент оригінального сигналу.

Результатом STFT є послідовність n-вимірних векторів, де n – це число "частотних комірок", що залежить від довжини перетворюваного сегмента сигналу. Застосування перетворення до довших сегментів дасть більше комірок, що збільшить роздільну здатність представлення сигналу у частотному просторі. Проте чим більший розмір сегментів, тим більші й затримка в обробці сигналу та часова роздільна здатність STFT, оскільки інформація про частоти вилучається з довших підпослідовностей сигналу. Вектори, що отримуються на останньому кроці алгоритму STFT, є коректним вкладенням вхідного сигналу.

Наступний відомий клас алгоритмів для побудови вкладень часових рядів базується на дублюванні вхідних даних з затримками. Зазвичай ці алгоритми застосовуються для реконструювання фазових просторів динамічних систем, але вони також є важливим інструментом для вилучення ознак [97], [98] та використовуються у добуванні даних [99]. Такі методи базуються на припущенні, що існує така двічі диференційовна *функція спостереження* $\alpha : \mathbb{R}^m \to \mathbb{R}$, що ставить у відповідність усім m-вимірним точкам відображення $f : \mathbb{R}^m \to \mathbb{R}^m$ дійсне число так, що можливо побудувати функцію вкладення у вигляді:

$$\begin{aligned} &\varphi(X) = \left(\alpha(X), \alpha(f(X)), \ldots, \alpha(f^{k-1}(X))\right), \\ &\alpha(f^i(X)) = x_{t-\tau_i}, \end{aligned} \tag{3.11}$$

де X – точка на атракторі системи у просторі $\mathbb{R}^m$, x_t – точка у часовому ряді з часовою міткою t, а τ_i – значення затримок, що позначають відстані у часі між поточною точкою та її сусідами у векторі вкладення.

Перш ніж побудувати вкладення часового ряду методом затримок, необхідно розв'язати дві проблеми. Перша – це вибір коректної розмірності для простору вкладення k, у якому, згідно з теоремою Такенса [35], динаміка реконструйованої траєкторії буде еквівалентна динаміці оригінальної системи. Друга

– підбір правильних значень затримок τ_i, що є особливо складно, оскільки теорема Такенса не дає жодних вказівок з цього приводу і в усіх доведеннях просто припускає, що вхідні дані не містять шуму, і покладає усі τ_i рівними одиниці.

Для розв'язання цих проблем існують роздільні та уніфіковані підходи. Роздільні методи спочатку обирають час затримки τ, а потім використовують його для оцінки оптимальної розмірності простору вкладення використовуючи одну з варіацій метода хибних ближніх сусідів [100]—[103]. Остаточний вибір розмірності зазвичай базується на оптимізації значення деякої статистики. Уніфіковані методи, усвідомлюючи існування тісного зв'язку між значеннями затримки та розмірності простору вкладення, застосовують підходи, що дозволяють підбирати значення τ_i динамічно кожен раз, коли вони збільшують розмірність вкладення на одиницю під час "циклів вкладення", таким чином оптимізуючи затримки та значення розмірностей одночасно. Яскравим прикладом уніфікованих алгоритмів є алгоритм PECUZAL [56], який використовує дві статистики для об'єктивної оцінки якості поточного вкладення, вирішує чи продовжувати збільшувати його розмірність, та обирає оптимальне значення затримки для кожного нового виміру.

Для вкладень з затримками також існують альтернативи, які застосовують складніші трансформації до часового ряду для отримання траєкторій у високовимірному просторі. Однією з таких альтернатив є метод Брумхеда-Кінга, що вилучає якісні динаміки з експериментальних даних [104]. Цей метод безпосередньо застосовує теорему Такенса до часового ряду і при цьому використовує мінімально можливе значення затримки $\tau = 1$ та достатньо високе значення розмірності простору вкладення d. Після цього до результату, представленого у вигляді матриці, застосовується сингулярне розкладення:

$$\frac{1}{\sqrt{N}} \begin{pmatrix} x_1 & x_2 & \dots & x_d \\ x_2 & x_3 & \dots & x_{d+1} \\ \vdots & \vdots & \ddots & \vdots \\ x_{N-d+1} & x_{N-d+2} & \dots & x_N \end{pmatrix} = U \cdot S \cdot V^T, \tag{3.12}$$

де N – це загальна довжина часового ряду, x_t – значення з часового ряду, що був відцентрований відносно осі t (тобто $\sum_{i=1}^{N} x_i = 0$), U – це матриця, рядки якої є векторами у просторі вкладення, а S – діагональна матриця, яка описує "важливість" кожного зі стовпців U, а V^T – ортогональна матриця, що змінює

базис масштабованих векторів вкладення так, щоб вони збігалися з вихідними даними. Оскільки за визначенням сингулярного розкладення матриця U є унітарною ($U^TU = I$), її стовпці формують ортонормальний базис, тобто компоненти високовимірної траєкторії, яку вона описує, не містять кореляцій та надлишкової інформації. Властивості матриці S також дозволяють оцінювати ступінь важливості кожного зі стовпців U та зменшувати розмірність, відкидаючи ті з них, яким відповідають значення S, менші заданого порогу.

Методи, описані вище, складають надійний набір інструментів для вкладення часових рядів, оцінки розмірності задачі та способів її зменшення. Щоб спростити подальше обговорення, введемо декілька означень та узагальнень.

По-перше, відзначимо, що визначення усіх описаних методів можна записати у вигляді множення матриць. STFT базується на дискретному перетворенні Фур'є, яке можна представити множенням вхідних даних на квадратну матрицю з комплексними елементами:

$$A = \frac{1}{\sqrt{N}} \begin{pmatrix} 1 & 1 & \dots & 1 \\ 1 & \omega & \dots & \omega^{N-1} \\ \vdots & \vdots & \ddots & \vdots \\ 1 & \omega^{N-1} & \dots & \omega^{(N-1)^2} \end{pmatrix}, \tag{3.13}$$

де $\omega = e^{\frac{i\pi}{N}}$.

Так само і методи побудови вкладення з затримкою можна звести до задачі вибору прямокутної матриці, після множення якої на сегмент часового ряду отримуємо реконструйований вектор у просторі вкладення:

$$A = \begin{pmatrix} a_{11} & a_{12} & \dots & a_{1d} \\ a_{21} & a_{22} & \dots & a_{2d} \\ \vdots & \vdots & \ddots & \vdots \\ a_{\tau_{max}1} & a_{\tau_{max}2} & \dots & a_{\tau_{max}d} \end{pmatrix}, \tag{3.14}$$

де стовпці A – одиничні вектори та $\forall j \exists i : a_{ij} = 1$, кількість стовпців d – розмірність простору вкладення, а кількість рядків τ_{max} – найбільше значення серед обраних затримок τ_i.

А щодо метода Брумхеда-Кінга, то його матрицю вкладення можна легко

вивести з означення (3.12):

$$A = (V^T)^{-1} \cdot S^{-1} = V \cdot S^{-1}. \tag{3.15}$$

Варто зазначити, що обидві матриці (3.14) та (3.15) описують внесок кожної точки часового ряду у вектор вкладення шляхом призначення їм деяких вагових коефіцієнтів. Різниця лише у тому, що матриця (3.14) жорстко обмежує кількість значень часового ряду, що можуть брати участь у формуванні кожної з компонент вектора вкладень, – вона відкидає усі значення, окрім одного. Матриця (3.15) набагато більш гнучка і визначає кожну компоненту вектора вкладень у вигляді лінійної комбінації з усіх точок поточного сегмента, що дуже схоже на те як оперує матриця для дискретного перетворення Фур'є (3.13).

По-друге, розглянемо рівняння (3.11) детальніше. Воно містить у собі припущення, що існує деяка невідома функція спостереження α, що ставить дійсні числа у відповідність високорозмірним точкам на атракторі. Визначення цієї функції ніяк не обмежує її конкретний вигляд окрім вимоги подвійної диференційовності. Однак на практиці вона майже завжди обирається як проста лінійна трансформація, яка проеціює одну з координат точок траєкторії оригінальної системи на множину дійсних чисел. З цього припущення також витікає рівність $\alpha(f^i(X)) = x_{t-\tau}$, яка у прикладних застосуваннях майже ніколи не справджується через наявність шуму в аналізованому сигналі та невідому структуру обох функцій f та α. Враховуючи (3.13), (3.14), і (3.15), другу частину (3.11) можна переписати як:

$$\alpha(f^i(X)) = \beta(x_t, x_{t-1}, \ldots, x_{t-l+1}), \tag{3.16}$$

де $\beta : \mathbb{R}^l \to \mathbb{R}$ – це *функція вилучення ознак*, що задає відображення сегментів часового ряду в область значень невідомої функції спостереження α.

З наведеного вище видно, що оптимальне вкладення може бути побудовано з використанням матриць типу (3.14), але властивості сингулярного розкладення показують, що не менш коректне вкладення можна отримати й за допомогою матриць (3.15). З цього випливає, що проблему побудови вкладення можна звести до задачі оптимізації рівності (3.16) відносно деякої функції втрат, розробленої навколо об'єктивної статистики для оцінювання якості вкладення в уніфікованих методах вкладення.

Оскільки обидві функції α та β невідомі, у цій ситуації доцільно використати

підхід до моделювання, орієнтований на дані. Якщо конкретніше, то можна використати спеціальну нейромережеву архітектуру – автокодувальник, що побудує двостороннє відображення між даними часового ряду та областями значень α і β. Головна ідея автокодуваьника у тому, щоб сполучити дві функції – кодувальник $\mathcal{E}$ і декодувальник $\mathcal{D}$ – так, щоб $x \approx \mathcal{D}(\mathcal{E}(x))$, з метою збільшення або зменшення розмірності x в області значень $\mathcal{E}$, яке називають *латентним простором*. Застосовуючи цю ідею в контексті рівнянь (3.11) і (3.13)-(3.15), отримуємо:

$$\begin{aligned} &\mathcal{E}(x_t, x_{t-1}, \ldots, x_{t-l}) = \beta(x_t, x_{t-1}, \ldots, x_{t-l}) = Y, \\ &\mathcal{D}(Y) = \alpha(f^{k-1}(X)) = x_t. \end{aligned} \tag{3.17}$$

Однією з переваг нейромережевого підходу перед іншими методами побудови вкладення, розглянутими в цьому розділі, є те, що структура функцій $\mathcal{E}$ і $\mathcal{D}$ не обмежена звичайними лінійними трансформаціями, і може містити будь-які нелінійності, індивідуально підібрані для кожного конкретного часового ряду.

Тепер розглянемо ефективність кожного з обговорених методів при роботі із синтетичними даними за критеріями, озвученими на початку розділу. Для таких порівнянь краще вибирати добре відому систему, щоб їх можна було з легкістю відтворити в різних обчислювальних середовищах, тому для цього тесту було обрано хаотичний режим системи Лоренца. Але беручи до уваги зауваження щодо структури функції спостереження α, її буде видозмінено, і замість того, щоб просто обирати одну координату з точок на атракторі, вона буде обчислювати їх норму. Для оцінки якості збереження характеристик траєкторії оригінальної системи у просторі вкладення, будуть порівняні спектри показників Ляпунова оригінальної системи та реконструйованого атрактора.

Усі обчислення та симуляції будуть виконані на мові Julia з використанням її різноманітних пакетів.

Система та функція спостереження, використані для генерування синтетичного часового ряду, мають наступний вигляд:

$$\begin{aligned} &\begin{cases} \dot{x} = 10(y - x) \\ \dot{y} = x(28 - z) - y \\ \dot{z} = xy - \dfrac{8}{3}z \end{cases}, \\ &\alpha(X) = ||X||_2. \end{aligned} \tag{3.18}$$

Часовий ряд, згенерований за допомогою (3.18) та відцентрований відносно осі t, зображено на Рис. 3.8.

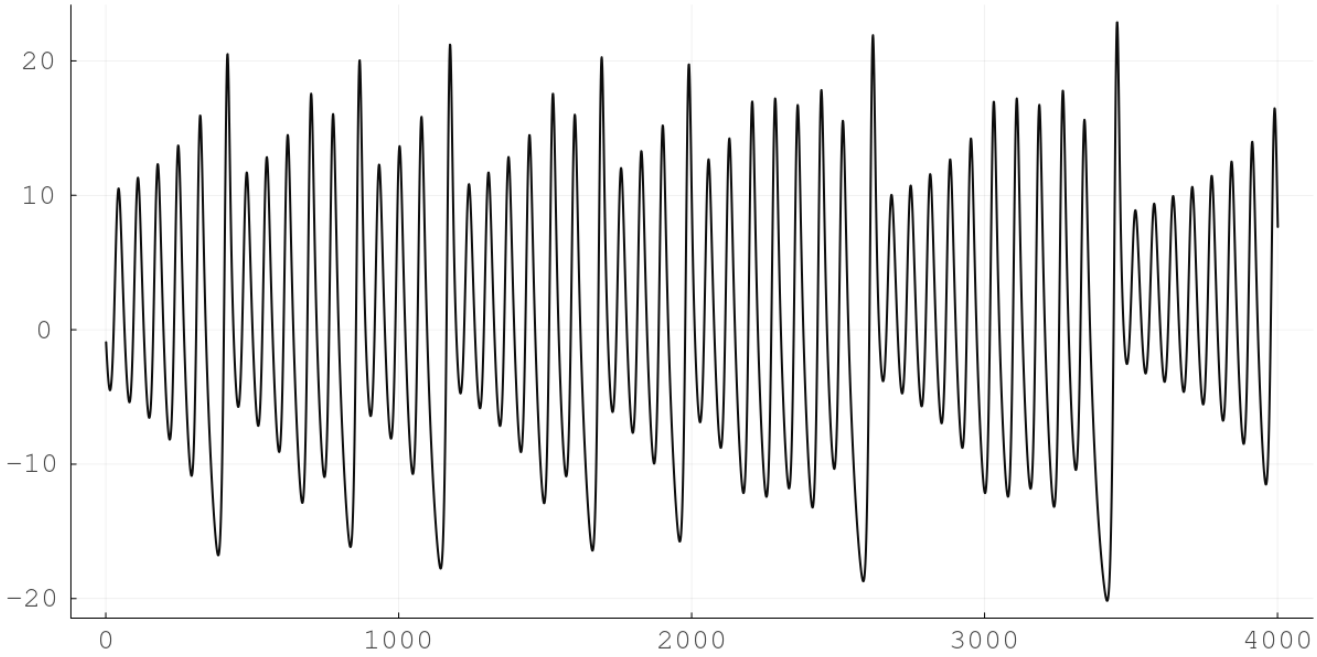


Рис. 3.8: Дані часового ряду, отримані з атрактора системи Лоренца з нелінійною функцією спостереження

Для отримання STFT вкладення тестового часового ряду, було обрано матрицю (3.13) розміром 10×10 та застосовано її до сегментів сигналу сумісної довжини. Суміжні сегменти мають 9 спільних семплів, тобто довжина стрибка мінімальна і рівна одиниці. Це перетворення відрізняється від інших тим, що його результат складається з комплексних чисел і його не можна безпосередньо візуалізувати. Для демонстрації буде візуалізовано лише дійсні частини отриманих векторів, а уявні відкинуто. Звичайний спосіб візуалізації результату STFT у вигляді спектрограми не використовується, щоб зберегти можливість порівняння отриманих векторів з результатом роботи інших методів вкладення.

Для побудови вкладення з затримкою було обрано уніфікований алгоритм PECUZAL. У цьому прикладі було використано реалізацію алгоритму з пакета DelayEmbeddings.jl [105]. Окрім самого часового ряду, алгоритму не було передано ніяких додаткових параметрів.

Для отримання вкладення за допомогою алгоритму Брумхеда-Кінга було використано реалізацію з пакета ChaosTools.jl [105]. Параметрами їй було передано сам часовий ряд, а також бажану розмірність простір вкладення 20. З результатів роботи алгоритму було побудовано матрицю (3.15), а усі її стов-

пці, окрім перших трьох, було відкинуто, щоб зберегти лише компоненти, яким відповідають найбільші значення елементів матриці S.

Нарешті, для побудови автокодувальника було обрано та оптимізовано дві прості лінійні моделі:

$$\begin{aligned} \mathcal{E}(x) &= A_e x + b_e = y \\ \mathcal{D}(y) &= A_d y + b_d \end{aligned}, \tag{3.19}$$

де $A_e \in \mathbb{R}^{N \times m}$ – матриця, що ставить у відповідність N суміжним точкам часового ряду m-вимірний вектор в латентному просторі, $A_d \in \mathbb{R}^{N \times m}$ – матриця, що перетворює вектор латентного простору на точку в часовому ряді, як показано в рівнянні (3.17), а b_e і b_d – вектори зсуву.

Траєкторії, що були отримані усіма обговореними методами вкладення, наведено на Рис. 3.9

Щоб порівняти якість кожного вкладення, було обчислено спектри показників Ляпунова для кожного з них і перевірено, наскільки сильно вони відрізняються від спектра оригінальної системи. Оскільки вкладені траєкторії не мають аналітичного представлення, було застосовано чисельний підхід до їх оцінки, а саме алгоритм з [106], що ґрунтується на QR декомпозиції, але замість використання аналітичного відображення, матриця локального якобіана будується чисельно для кожної точки з використанням інформації про її сусідів за алгоритмом, описаним в [107]. Значення обчислених спектрів показників Ляпунова наведено в Табл. 3.2.

Ще одна міра якості вкладення – це L статистика, якою користується алгоритм PECUZAL для оцінки поточної якості реконструкції траєкторії між циклами вкладень. Чим вище значення статистики, тим менша кореляція між компонентами вектора вкладень. Значення L статистики для траєкторії оригінальної системи та для кожного з побудованих вкладень також наведені у Табл. 3.2.

	Істинні значення	STFT	PECUZAL	Брумхед-Кінг	Автокодувальник
λ_1	0.91	3.9	2.18	0.04	1.9
λ_2	0	1.08	0.5	0	0.22
λ_3	-14.57	-28.02	-13	-0.19	-12.07
L	-0.09	1.41	0.43	-0.48	0.4

Табл. 3.2: Значення спектрів показників Ляпунова та L статистика для хаотичного атрактора системи Лоренца та для його реконструйованих траєкторій.

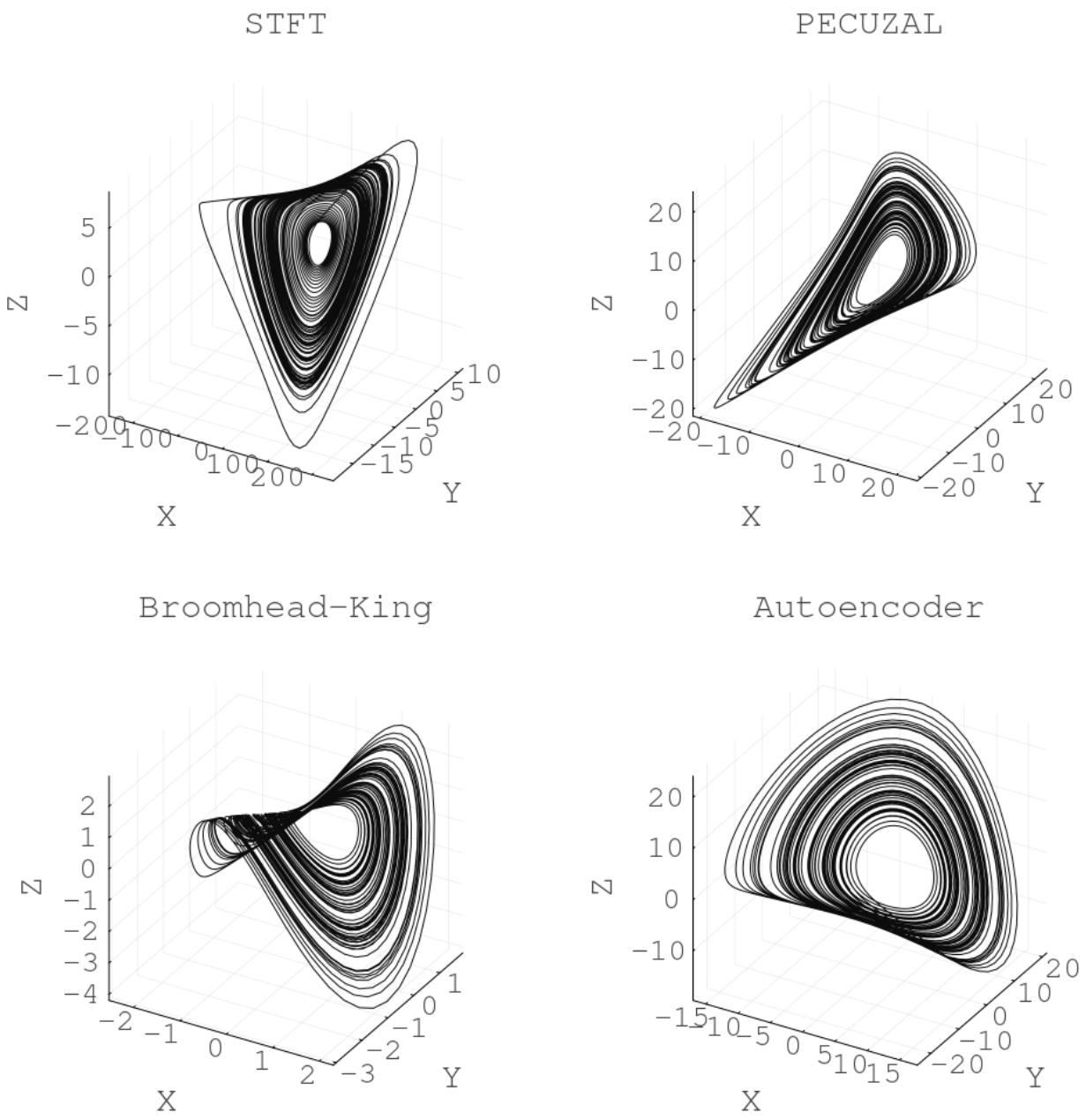


Рис. 3.9: Реконструкції тривимірного атрактора системи Лоренца, отримані різними методами вкладення.

З таблиці 3.2 видно, що метод PECUZAL та нейромережа-автокодувальник найкраще впоралися з побудовою якісної реконструкції та збереженням динамічних характеристик траєкторії системи Лоренца.

3.5 hNODE модель для параметричного генерування голосового сигналу

Моделювання голосового сигналу мало чим відрізняється від моделювання звичайних періодичних сигналів. Подібне застосування моделі hNODE вже було розглянуто в попередніх розділах, де вона показала гарні результати, а тому для розв'язання даної задачі побудуємо модель саме на базі архітектури hNODE.

Для побудови моделі необхідно розв'язати наступні задачі:

1. Коректно розбити вхідний сигнал на сегменти та розробити метод попередньої підготовки кожного сегмента до аналізу.
2. Обрати архітектуру для кодувальника і декодувальника.
3. Обрати архітектури для вкладеної NODE та моделі для генерування її параметрів.
4. Розробити коректну функцію втрат для тренування моделі та підготувати навчальні дані.

Розглянемо детально розв'язання кожної з цих задач.

3.5.1 Розбиття вхідного сигналу на сегменти та попередня обробка сегментів

Голосовий сигнал можна розглядати як динамічний процес, якісні характеристики якого постійно змінюються у процесі мовлення. З цього випливає, що модель для його описання має постійно адаптувати власні параметри, щоб підтримувати поведінку своїх траєкторій відповідною до поточних характеристик голосу. На практиці це означає, що голосовий сигнал можна розбити на сегменти, для кожного з яких і проводити оцінку параметрів системи, що його моделює. Таким чином отримаємо серію наборів параметрів для суміжних сегментів сигналу, що описують зміну якісних характеристик системи.

Розглянемо часовий ряд $U = (u_1, u_2, \ldots, u_n)$, заданий дискретним сигналом $u(n)$ з часовим кроком Δt. Сегментом $S_{i,m}$ довжиною $m \leq n - i + 1$ часового ряду будемо називати підпослідовність $(u_i, u_{i+1}, \ldots, u_{i+m-1}) \subset U$.

Задача розбиття вхідного сигналу U на сегменти $S_{i,m}$ є тривіальною та зводиться до вибору параметра сегментації m. Для даної задачі довжина сегмента має задовольняти модель-кодувальник та модель-контролер. Користуючись попереднім знанням про об'єкт досліджень, оберемо значення $m = 3f_s F_{\min}^{-1}$, де f_s – частота дискретизації, а $F_{\min}$ – мінімальна частота періодичного сигналу, який система має бути здатна обробити.

Значення $F_{\min}$ оберемо так, щоб вона покривала найнижчу доступну людині ноту – "мі" великої октави (E_2), частота якої становить приблизно 80 Герців. Проте враховуючи необхідність застосування фільтрів низьких частот на етапі попередньої обробки, покладемо $F_{\min} = 60$ Гц, щоб запобігти спектральному спотворенню в околі E_2.

Значення f_s підібрати складніше, оскільки воно може бути різним у різних середовищах, або навіть в одному і тому самому середовищі. Для того, щоб забезпечити коректну роботу моделі за будь-яких значень f_s, необхідно зафіксувати одне значення частоти дискретизації, що задовольняє потребам моделі, та проводити редискретизацію усіх сигналів, чия попередня частота дискретизації є відмінною від бажаної. У даному контексті бажана частота є такою, що забезпечує збереження найвищої доступної людині ноти – "до" третьої октави (C_6), частота якої приблизно 1046 Гц, а також її гармонік для уможливлення коректного спектрального та кепстрального аналізу мовлення з такою високою частотою F_0. Таким чином, оберемо бажаною частотою дискретизації моделі $f_s^M = 16000$ герців – її частота Найквіста F_{Nyquist}, що становить $f_s/2 = 8000$ герців, дозволить зберегти не тільки частоти в околицях границі голосового діапазону людей, але і як мінімум сім гармонік.

Отже, результатом попередньої обробки сигналу має бути послідовність суміжних сегментів редискретизованого сигналу з цільовою частотою $f_s^M = 16000$ герців з $m = 3f_s^M F_{\min}^{-1} = 800$ семплів у кожному. При цьому спектр кожного з сегментів не має містити компонентів, що лежать поза діапазоном допустимих частот $[F_{\min}, F_{\max}]$.

Наївний підхід до реалізації такої обробки ґрунтувався б на незалежній обробці кожного сегмента $S_{i,m}$. При такому підході сегменти б обиралися безпосередньо з сигналу U та мали б довжину, з якою в результаті редискретизації було отримано новий сегмент довжиною $m = 800$ (наприклад, для вхідних сигналів з частотою дискретизації 44.1 кГц, довжина необробленого сегмента б

становила $3 \cdot 44100 \cdot 60^{-1} = 2205$), а фільтр небажаних частот застосовувався б на кожному сегменті окремо. До недоліків такого підходу також належить те, що між суміжними сегментами можуть виникнути розриви, оскільки будь-яке фільтрування може призвести до спотворень сигналу, що значно погіршать його звукові характеристики.

Врахувавши особливості предметної області, розробимо новий підхід, заснований на потоковій обробці. По-перше, зауважимо, що сегменти мають бути суміжними, а вкладена модель NODE може продукувати траєкторії, довжина яких менша за m, перш ніж повторно виконувати адаптацію параметрів. Це означає, що новий сегмент можна формувати динамічно, семпл за семплом, а попередні семпли брати з попереднього сегмента.

Такий підхід до обробки можна реалізувати у вигляді FIR фільтру зі змінним внутрішнім станом. Використаємо реалізацію такого FIR фільтру із пакета для роботи з цифровими сигналами DSP.jl. Вона має внутрішній буфер сталої довжини, до якого записуються результати застосування нерекурсивного фільтра типу (1.55) до вхідного сигналу. Інтерфейс FIR фільтрів, реалізований в DSP.jl також дозволяє задавати коефіцієнт для інтерполяції або децимації вхідного сигналу. Цей коефіцієнт задається відношенням бажаної частоти дискретизації до фактичної – якщо бажана частота менша за фактичну, фільтр буде пропускати деякі семпли сигналу перш ніж записувати їх у внутрішній буфер, а якщо фактична менша за бажану, то виконувати інтерполяцію між відомими суміжними семплами. Таким чином, система попередньої обробки, побудована на базі FIR фільтру, буде використовувати значно менше пам'яті у порівнянні з наївним підходом та забезпечить відсутність небажаних спотворень при конкатенації суміжних сегментів, оскільки "сегмент" завжди один і фільтру завжди доступна інформація про попередні семпли.

Єдиним недоліком використання FIR фільтру є його непристосованість до використання в нейромережевих моделях, а точніше – в моделях, побудованих інструментами із фреймворку Lux.jl [108]. Цей фреймворк вимагає, щоб усі параметри та стани були незмінними під час обчислення моделі, а сама модель має реалізовувати один з абстрактних інтерфейсів, описаних фреймворком. Напишемо модель-обгортку, що дозволяє використовувати FIR фільтри в моделях, побудованих за допомогою Lux.jl.

Оберемо базовий інтерфейс, який буде реалізовувати нова модель. Lux.jl має

декілька абстрактних інтерфейсів, які можна реалізувати для створення нових моделей. Найпростіший інтерфейс – `AbstractLuxLayer`, який є базовим для усіх моделей в Lux.jl та робить найменше припущень щодо поведінки моделі, що його реалізує. Проте враховуючи специфіку роботи FIR фільтра, доцільніше буде використати інтерфейс `AbstractRecurrentCell`, оскільки він дасть змогу фільтрові використовувати один і той самий буфер для обробки послідовних входів, що дозволить зекономити пам'ять та спростити роботу з самими буферами. Визначимо структуру нової моделі мовою Julia:

```
1  struct StreamFilter <: Lux.AbstractRecurrentCell
2      filter::FIRFilter
3      n::Integer
4
5      function StreamFilter(
6          n::Int,
7          sample_rate::Int;
8          target_sample_rate::Int=sample_rate,
9          f0_floor::Float32,
10         f0_ceil::Float32,
11         filter_length::Int=Int(sample_rate f0_floor),
12         window=hanning
13     )
14         h = digitalfilter(
15             DSP.Bandpass(f0_floor, f0_ceil; fs=sample_rate),
16             FIRWindow(window(filter_length))
17         )
18         fir_filter = FIRFilter(
19             h,
20             target_sample_rate // sample_rate
21         )
22         return new(fir_filter, n)
23     end
24 end
```

Модель називається StreamFilter та реалізує `AbstractRecurrentCell`. Щоб створити модель, необхідно викликати її конструктор з параметрами:

1. `n` – ціле число, що визначає розмір сегмента обробленого сигналу, який буде повертатися моделлю;

2. `sample_rate` – ціле число, що задає частоту дискретизації вхідного сигналу;

3. `target_sample_rate` – ціле число, що задає цільову частоту дискретизації;

4. `f0_floor` – число з рухомою комою, що задає значення мінімальної допустимої частоти $F_{\min}$;

5. `f0_ceil` – число з рухомою комою, що задає значення максимально допустимої частоти $F_{\max}$;

6. `filter_length` – ціле число, що визначає кількість коефіцієнтів FIR фільтра (його довжину); за замовчанням дорівнює $f_s^M / F_{\min}$;

7. `window` – посилання на віконну функцію, що буде використовуватися для обчислення коефіцієнтів фільтра; за замовчанням використовується функція Ханна.

Обчислення моделі виконується наступним чином:

```
function (sf::StreamFilter)(
    xs::AbstractMatrix,
    ps,
    st::NamedTuple
)
    reset!(sf.filter)

    m = size(xs)[2]
    buffers = zeros(Float32, sf.n, m)
    filters = [deepcopy(sf.filter) for i in 1:m]

    return sf((xs, (buffers, filters)), ps, st)
end

```

```
15 function (sf::StreamFilter)(
16     (xs, (buffers, filters))::Tuple{<:AbstractMatrix, Tuple
       },
17     ps,
18     st::NamedTuple
19 )
20     batches = zip(eachcol(xs), filters, eachcol(buffers))
21     for (x, filter, buffer) in batches
22         y = sf.gain * DSP.filt(filter, sf.pregain * x)
23         shiftin!(buffer, y)
24     end
25     return (copy(buffers), (buffers, filters)), st
26 end
```

Перша функція, що окрім параметрів і стану приймає лише вхідні сигнали у вигляді матриці, виконує скидання стану вкладених фільтрів та ініціює буфери, куди будуть записуватись результати їх застосування. Наприклад, якщо модель було проініційовано з параметром $n = 800$, а розмірність вхідної матриці xs становить 40×10, то буде створено 10 фільтрів та 10 буферів довжиною 800 семплів.

Друга функція містить безпосередньо логіку застосування фільтрів та оновлення буферів. Вона приймає як параметр кортеж із матриці вхідних сигналів та кортежу, що містить проініційовані першою функцією фільтри та буфери. Застосування фільтрів відбувається для кожної колонки матриці xs, а результат заштовхується у відповідні буфери з кінця. Наприклад, якщо коефіцієнт інтерполяції/децимації фільтрів становить 0.5, довжини буферів – 800, а розмірність матриці входів xs – 40×10, то в кожен з десяти буферів буде заштовхано по 20 семплів обробленого сигналу (усі елементи усіх буферів буде зсунуто вліво на 20 елементів, а у вільне місце справа додадуться нові елементи). Детальний алгоритм процесу:

1. Отримати k фрагментів вхідних сигналів довжиною n (матриця xs розмірністю $n \times k$), k FIR фільтрів (масив `filters`), та k буферів довжиною m (матриця ys розмірністю $m \times k$).

2. Застосувати кожен з фільтрів до фрагмента відповідного сигналу, в ре-

зультаті отримавши нову матрицю xs' розмірністю $l \times k$.

3. Використати значення з xs' для оновлення буферів ys таким чином, що

$$ys'_{ij} = \begin{cases} xs'_{(i-m)j} & \text{, if } i > m, \\ ys_{(i+m)j} & \text{, otherwise.} \end{cases}$$

4. Повернути кортеж $(\text{copy}(ys'), (ys', \texttt{filters}))$.

Нова модель StreamFilter є повністю сумісною з іншими інструментами фреймворка Lux.jl (окрім автоматичного диференціювання) та може використовуватися для побудови будь-яких моделей. Важливою властивістю цієї моделі є те, що її можна використовувати разом з моделлю `StatefulRecurrentCell`, яка перетворює вкладену в неї рекурентну модель на звичайну функцію з одним входом й одним виходом, а значення, необхідні для її рекурентної роботи, підтримуються в об'єкті стану моделі `st`. Це дозволяє включати StreamFilter до вже існуючих моделей без необхідності змінювати їх код.

Отже, маємо модель, яка одночасно розв'язує як задачу розбиття сигналу на сегменти, так і задачу його попередньої обробки. При цьому забезпечується ефективне використання пам'яті, відсутність спотворень при на стиках суміжних сегментів, та можливість динамічно змінювати відстань між ними, варіюючи перший вимір вхідної матриці xs.

3.5.2 Вибір архітектури кодувальника та декодувальника

Як було продемонстровано у розділі 3.4, нейромережа-автокодувальник є ефективним способом перетворення даних часового ряду в точки латентного простору та навпаки – асоціювання значень часового ряду з латентними точками. Тому застосуємо такий підхід і в даній задачі.

Спершу сформуємо архітектуру моделі-кодувальника.

Входами цієї моделі будуть сегменти, отримані на етапі сегментації та попередньої підготовки даних. Проте довжина цих сегментів містить у собі три періоди мінімальної частоти, що забагато для побудови навіть звичайного вкладення з затримкою. Можемо це перевірити, використавши алгоритм вкладення PECUZAL на записах голосу з каталогу LibriSpeech. Алгоритм повернув насту-

пні значення затримок τ:

$$\begin{bmatrix} 0 & 9 & 5 & 7 & 43 & 35 & 8 & 39 & 41 & 17 \end{bmatrix}$$

Отже, розмірність вхідних даних можна суттєво обмежити, значно зменшивши кількість параметрів моделі. Оберемо максимальну довжину вхідних даних як $N = f_s^M / F_{\min} = 266$, щоб дозволити моделі брати до уваги повний період будь-якої частоти, вищої за мінімальну.

Не зважаючи на те, що вхідні дані не містять частоти поза діапазоном $[F_{\min}, F_{\max}]$, сигнал все ще може містити різноманітні небажані домішки. Щоб надати моделі змогу позбутися цих шумів, зробимо першими шарами послідовність згорткових нейромереж з тотожною функцією активації, оскільки їх визначення майже еквівалентне формі FIR фільтрів, а властивість таких мереж "віднаходити структури" у вхідних даних дозволить решті шарів кодувальника працювати лише з "важливими" спектральними компонентами отриманих сегментів. Використаємо два згорткові шари з числом коефіцієнтів $k = f_s^M / F_{\max} = 26$. Два таких послідовні шари, застосовані до вхідних даних довжиною $N = 266$, дадуть новий відфільтрований сегмент довжиною $N' = 216$.

До результату роботи згорткових мереж вже можна застосувати звичайний багатошаровий персептрон, який поставить йому у відповідність точку в десятивимірному латентному просторі. Його побудуємо у вигляді наступної двошарової моделі:

$$f(x) = W_2 \tanh(W_1 x + b_1) + b_2,$$

де $W_1 \in \mathbb{R}^{216\times216}$, $b_1 \in \mathbb{R}^{216}$, $W_2 \in \mathbb{R}^{10\times216}$, $b_2 \in \mathbb{R}^{10}$.

Для побудови декодувальника можна застосувати звичайну одношарову нейромережу з тотожним відображенням у якості функції активації. Проте для того, щоб зробити модель більш гнучкою та дозволити їй моделювати нелінійні зв'язки між змінними вкладеної NODE, додамо також залишковий блок у вигляді:

$$y_1 = x + tanh(W_2 tanh(W_1 x + b_1) + b_2)$$

Остаточний результат буде отриманий як зважена сума компонентів y_1 зі зсувом $y = W y_1 + b$, $y \in \mathbb{R}$.

3.5.3 Вибір архітектури для вкладеної NODE та моделі для генерування її параметрів

Спираючись на результати, отримані в розділі 2.6.4, побудуємо модель NODE на базі наступної системи ЗДР:

$$\dot{\mathbf{x}}(t) = A\mathbf{x}(t) + \mathbf{c} + \sigma(W\mathbf{x}(t) + \mathbf{b}), \tag{3.20}$$

де $A, W \in \mathbb{R}^{10\times 10}$, $\mathbf{c}, \mathbf{b} \in \mathbb{R}^{10}$, та функція σ – степенева функція активації (2.21) з фіксованими параметрами $\alpha = \beta = c = 0.5$. Параметри функції активації обрані таким чином, щоб отримати переваги необмеженої неперервно диференційовної функції під час навчання моделі, а також забезпечити ефективне обчислення траєкторій, що буде покладатися на оптимізовані алгоритми отримання квадратного кореня числа у сучасних комп'ютерах.

Модель hNODE здатна генерувати параметри для вкладеної NODE з довільного контексту за допомогою моделі-контролера. Отже, замість того, щоб використовувати контролер безпосередньо на вхідних даних (фрагменті голосового сигналу), їй на вхід можна подавати легко інтерпретовані значення, що описують поточний сегмент. Такими значеннями є описані вище фундаментальна частота, спектральні характеристики, та гучність. Для вилучення частоти та спектральної оболонки можна використати відповідно алгоритми JDIO та JCheapTrick, а для оцінки гучності використати корінь середнього квадратичного поточного сегмента сигналу.

Проте у такого підходу є істотний недолік. Усі ці значення належать до абсолютно різних предметних областей та мають різні діапазони значень і розподіли. Оцінка фундаментальної частоти являє собою вектор з двома значеннями – найбільш ймовірна частота в герцах та коефіцієнт впевненості в діапазоні $[0, 1]$. Спектральна характеристика містить $O(f_s^M)$ елементів, що не обмежені ні зверху, ні знизу. Оцінка гучності задається одним єдиним скаляром. Для подолання цього недоліку застосуємо метод для зменшення кількості елементів спектра та створимо нову модель, що дозволить уніфікувати різнородні дані в один вектор характеристик.

Залежно від налаштувань JCheapTrick, довжина спектра може сягати значення частоти дискретизації. Але така велика кількість інформації не обов'язкова для наших потреб. В галузі аналізу мовлення широко використовується ефе-

ктивний метод стиснення інформації про спектральні характеристики голосу – перетворення елементів спектра на *мелчастотні кепстральні коефіцієнти* (mel-frequency cepstral coeffitients, MFCC) [109]—[111].

Перший крок цього підходу полягає у переході від герців до мелів – суб'єктивної шкали частот, відліки якої на слух сприймаються як рівновіддалені одне від одного [112]. Перехід від герців до мелів здійснюється за формулою [113]:

$$\text{mel} = 2595 \log_{10} \left(1 + \frac{f}{700}\right). \tag{3.21}$$

Логарифм, присутній у формулі (3.21) означає, що високі частоти, близькі одне до одного у герцах, будуть значно ближчими у мелах. Дійсно, "відстань" між частотами 440 Гц (≈ 549.63 мел) й 880 Гц (≈ 917.48 мел) на слух сприймаються так само як і відстань між 1760 Гц (≈ 1416.45 мел) та 3520 Гц (≈ 2024.65 мел) і становить одну октаву. В мелах ці дві відстані будуть значно ближчі одне до одного, ніж у герцах.

Другий крок цього підходу полягає у підрахуванні k зважених сум, що дадуть грубе наближення оригінального спектра і значно зменшують його розмірність. На практиці перший та другий крок виконують одночасно за допомогою так званого "банку фільтрів". Коефіцієнти для зважених сум обчислюються так, щоб вони одразу містили інформацію про відповідність між герцами та мелами. Приклад такого банку фільтрів наведено на Рис. 3.10.

Швидке обчислення зважених сум досягається шляхом формування матриці з їх вагових коефіцієнтів, та множенням цієї матриці на спектр сегмента, в результаті чого отримується вектор довжиною k.

Останнім кроком побудови є обчислення кепстра отриманої грубої апроксимації вихідного спектра. Але у цьому підході зазвичай застосовують не перетворення Фур'є, а дискретне косинусне перетворення, оскільки його результат містить лише дійсні числа, якими значно легше маніпулювати в практичних застосунках.

У даній задачі покладемо $k = 29$, що значно зменшує розмірність вхідних даних, але ціною втрати точності та інтерпретовності цього джерела даних про вхідний сигнал.

Тепер розглянемо нову модель для поєднання різнородних даних в один вектор. Вона є простим узагальненням звичайної одношарової нейромережевої

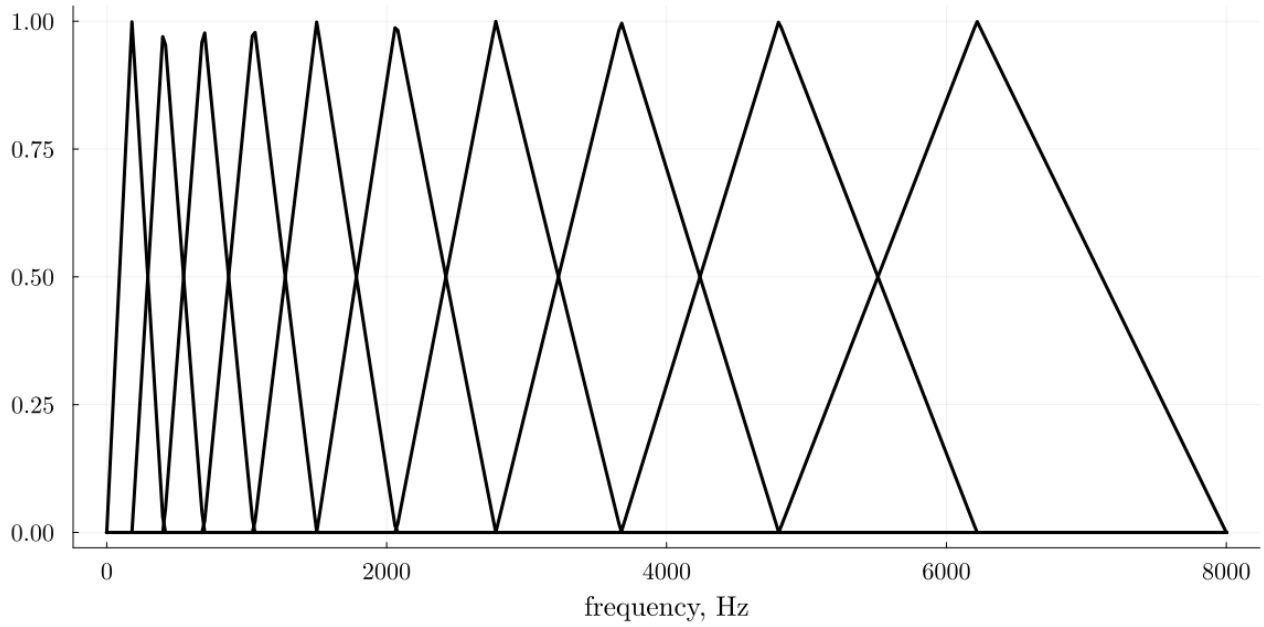


Рис. 3.10: Банк фільтрів з десяти зважених сум для сегмента довжиною $n = 800$ і частотою дискретизації $f_s = 16$ кГц.

моделі, що приймає на вхід не один, а декілька векторів, паралельно застосовує для кожного з них перетворення, задані незалежними одношаровими нейромережами, що повертають вектори однакової довжини, та об'єднує отримані вектори у фінальний результат, керуючись деякою функцією. Код моделі мовою Julia має наступний вигляд:

```
struct MergeLayer <: Lux.AbstractLuxLayer
    ins::Tuple
    out::Int
    connection
    activation::Function
    function MergeLayer(
        io,
        connection=+,
        activation=identity
    )
        ins, out = io
        new(ins, out, connection, activation)
    end

```

```
15 end
16
17 function (f::MergeLayer)(us, ps, st)
18     Ws = ps[:Ws]
19     bs = ps[:bs]
20
21     W_keys = keys(Ws)
22     b_keys = keys(bs)
23
24     results = [
25         f.activation.(Ws[W_key]*u .+ bs[b_key])
26         for (u, W_key, b_key) in zip(us, W_keys, b_keys)
27     ]
28
29     connected_results = reduce(f.connection, results)
30
31     return connected_results, st
32 end
33
34 Lux.initialparameters(rng::AbstractRNG, f::MergeLayer) = (
35     Ws = (; [Symbol("W_$i") => rand(rng, Float32, f.out, f.
        ins[i]) .- 0.5f0 for i in 1:length(f.ins)]...),
36     bs = (; [Symbol("b_$i") => zeros32(f.out) for i in 1:
        length(f.ins)]...),
37 )
38
39 Lux.initialstates(rng::AbstractRNG, f::MergeLayer) =
     NamedTuple()
```

Дана реалізація покладається на динамічне створення іменованих кортежів з їх подальшим використанням в обчисленнях. Наприклад, якщо модель було проініційовано з параметрами `io = (2, 1, 29)` $\to$ `880`, `connection = +`, `activation = tanh`, то в параметрах моделі буде створено три матриці $W_1 \in \mathbb{R}^{880\times 2}$, $W_2 \in \mathbb{R}^{880\times 1}$ і $W_3 \in \mathbb{R}^{880\times 29}$, а також три вектори $b_1, b_2, b_3 \in \mathbb{R}^{880}$. Кожен вхідний вектор буде використано для обчислення відповідної одношарової

нейромережі $y_i = \tanh(W_i x_i + b_i)$, а остаточним результатом буде сума усіх підрезультатів $\sum_{i=1}^{3} y_i$. У підсумку отримаємо вектор $y \in \mathbb{R}^{880}$.

До цього вектора y потім застосовуємо двошаровий персептрон та отримуємо вектор параметрів для вкладеної моделі NODE довжиною 220.

3.5.4 Функція втрат для тренування моделі та підготовка навчальних даних

Функція втрат має брати до уваги усі важливі критерії оцінки моделі.

Найбільш очевидним критерієм є те, що декодовані траєкторії, породжені нею, мають збігатися із модельованим голосовим сигналом. Відповідності моделі цьому критерію можна сприяти, використавши просту функцію, що повертає середньоквадратичне відхилення результату її роботи від бажаної траєкторії:

$$\mathcal{L} = \frac{1}{N} \sum_{i=1}^{N} (y_i - \hat{y}_i)^2, \tag{3.22}$$

де N – довжина згенерованої траєкторії.

Враховуючи специфіку предметної області, зауважимо, що усі цільові значення траєкторій існують у вузькому діапазоні $-1 \leq y \leq 1$, $\forall y$. Це призводить до того, що значення різниці між бажаним і фактичним результатом буде зменшуватись, якщо її піднести до квадрата. Тому введемо коефіцієнт масштабування $C \gg 0$ та включимо його до функції втрат (3.22):

$$\mathcal{L} = \frac{1}{N} \sum_{i=1}^{N} (C(y_i - \hat{y}_i))^2. \tag{3.23}$$

Функція втрат (3.23) буде добре працювати лише на простих періодичних сигналах, що не містять багато гармонік. Тому введемо другий критерій, який дозволить моделі правильно ідентифікувати та реплікувати гармонічний склад модельованого сигналу. На практиці цей критерій буде полягати в обчисленні спектрів згенерованих траєкторій та підрахунку суми квадратів різниць значень густини в кожній парі частотних комірок. Таким чином, функція втрат набуде вигляду:

$$\begin{aligned}\mathcal{L} = &\frac{1}{N}\sum_{i=1}^{N}(C(y_i - \hat{y}_i))^2 \\ &+ \frac{2}{N}\sum_{i=1}^{N/2}(\log_{10}(|U_{\text{DFT}}(i)|) - \log_{10}(|\hat{U}_{DFT}(i)|))^2.\end{aligned} \tag{3.24}$$

Ще одним критерієм моделі є те, що вкладена в неї пара кодувальник-декодувальник має коректно працювати з усіма можливими входами та виходами моделі. Наразі функція втрат перевіряє правильність роботи кодувальника лише на одній точці – початком згенерованої траєкторії. Якщо під час генерації траєкторії довжиною $N = km$ модель змінила параметри k разів для генерування сегментів довжиною m, то кодувальник буде викликано лише один раз на початку траєкторії, через що функція втрат не зможе оцінити його роботу у точках $i_0 + (k-1)m$, $k > 1$. Для цього введемо ще один критерій, який буде окремо підраховувати втрати застосування пари кодувальник-декодувальник:

$$\begin{aligned}\mathcal{L} = &\frac{1}{N}\sum_{i=1}^{N}(C_1(y_i - \hat{y}_i))^2 \\ &+ \frac{2}{N}\sum_{i=1}^{N/2}(\log_{10}(|U_{\text{DFT}}(i)|) - \log_{10}(|\hat{U}_{DFT}(i)|))^2 \\ &+ \frac{m}{N}\sum_{k=1}^{N/m}(C_2(y_{(k-1)m+1} - \mathcal{D}(\mathcal{E}(y_{(k-1)m+1}))))^2,\end{aligned} \tag{3.25}$$

де $k = N/m$ – кількість разів, коли модель створювала нові параметри для генерації нового сегмента траєкторії довжиною m.

Функція втрат (3.25) враховує усі критерії, що стосуються предметної області. Проте існують ще мета критерії, які дозволять моделі ефективніше узагальнювати та мати більш розріджені параметри.

Для покращення узагальнення широко використовують різні типи "регуляризації". Регуляризація – це додатковий критерій, що включається до функції втрат з метою покращення узагальнювальних властивостей нейромережі та запобігання проблеми перенавчання. Цей критерій зазвичай залежить від значень

параметрів самої моделі [114]—[116], але також може брати до уваги й значення її проміжних результатів [117]. Оскільки модель hNODE має два набори параметрів – безпосередні параметри моделі Θ та згенеровані параметри $\mathcal{H}(\Theta)$ для вкладеної NODE – складемо регуляризацію, що мінімізує суму абсолютних значень та квадратів обох наборів параметрів. Регуляризація Θ буде запобігати перенавчанню моделі та сприятиме її шумостійкості. Регуляризація параметрів $\mathcal{H}(\Theta)$ дозволить моделі генерувати менш екстремальні значення параметрів та сприяти утворенню більш компактного підпростору, у якому вони існують:

$$\begin{aligned}\mathcal{L} = &\frac{1}{N}\sum_{i=1}^{N}(C_1(y_i - \hat{y}_i))^2 \\ &+ \frac{2}{N}\sum_{i=1}^{N/2}(\log_{10}(|U_{\text{DFT}}(i)|) - \log_{10}(|\hat{U}_{DFT}(i)|))^2 \\ &+ \frac{m}{N}\sum_{k=1}^{N/m}(C_2(y_{(k-1)m+1} - \mathcal{D}(\mathcal{E}(y_{(k-1)m+1}))))^2 \\ &+ \frac{1}{\ell(\Theta)}\sum_{i=1}^{\ell(\Theta)}|\Theta_i| + \frac{1}{\ell(\mathcal{H}(\Theta))}\sum_{i=1}^{\ell(\mathcal{H}(\Theta))}(|\mathcal{H}(\Theta)_i| + \mathcal{H}(\Theta)_i^2).\end{aligned} \tag{3.26}$$

Функція (3.26) містить усі необхідні критерії для навчання моделі hNODE в обраній предметній області, а також включає компоненти, що сприяють її шумостійкості та запобігають перенавчанню. Проте вона не бере до уваги структуру вхідних даних та їх кількість.

У фреймворку Lux.jl, усі обчислення проводяться над багатовимірними масивами. Останній вимір цього масиву позначає кількість "партій" (batches) даних, тобто кількість унікальних входів та очікуваних виходів, які модель має обчислити одночасно. Наприклад, для простої моделі $f(x) = Wx + b$, вхід x може бути як вектором-стовпцем, так і матрицею з великою кількістю стовпців. У цьому випадку кількість стовпців в x і буде кількістю партій, а вектор зсуву b додається до кожного рядка матриці Wx окремо. На практиці це означає, що функцію втрат для моделі можна обчислювати "паралельно" одразу для великої кількості даних і отримати значення градієнта для них усіх одночасно. Такий підхід дозволяє більш ефективно використовувати ресурси комп'ютера, що виконує обчислення, а також зменшити вплив шуму в кожній окремій парі

вхід-вихід на результуючий градієнт.

Якщо значно збільшити кількість партій у функції (3.26), то компоненти, введені у (3.25), почнуть домінувати над рештою критеріїв регуляризації. Щоб запобігти цьому, введемо усереднення, яке дозволить функції втрат бути незалежною від кількості партій:

$$\begin{aligned}\mathcal{L} = &\frac{1}{BN}\sum_{i=1}^{N}(C_1(y_i - \hat{y}_i))^2 \\ &+ \frac{2}{BN}\sum_{i=1}^{N/2}(\log_{10}(|U_{\mathrm{DFT}}(i)|) - \log_{10}(|\hat{U}_{DFT}(i)|))^2 \\ &+ \frac{m}{BN}\sum_{k=1}^{N/m}(C_2(y_{(k-1)m+1} - \mathcal{D}(\mathcal{E}(y_{(k-1)m+1}))))^2 \\ &+ \frac{1}{\ell(\Theta)}\sum_{i=1}^{\ell(\Theta)}|\Theta_i| + \frac{1}{\ell(\mathcal{H}(\Theta))}\sum_{i=1}^{\ell(\mathcal{H}(\Theta))}(|\mathcal{H}(\Theta)_i| + \mathcal{H}(\Theta)_i^2),\end{aligned} \tag{3.27}$$

де B – кількість партій навчальних даних.

Як навчальні дані було обрано комбінацію синтезованих періодичних сигналів і аудіозаписів з датасета LibriSpeech.

Оскільки модель, описана в розділі 3.5.1 не містить жодних параметрів, розбиття сигналу, його попередня обробка, а також виділення визначних характеристик за допомогою алгоритмів, описаних в розділах 3.1 і 3.2, можемо виконати на етапі підготовки тренувальних даних. Проте насправді головною причиною цього кроку є те, що реалізація автоматичного диференціювання Zygote.jl [118], що використовуються фреймворком Lux.jl під час тренування моделей, не здатні працювати з функціями пакета DSP.jl.

Для того, щоб модель могла працювати з частково обробленими даними, розділимо її обчислення на три етапи:

1. визначення початкової точки для інтегрування,
2. обробка вхідного сигналу для та виділення визначних характеристик,
3. визначення параметрів вкладеної NODE з отриманих характеристик та обчислення цільової траєкторії.

В коді ці три етапи реалізуються як три окремі функції, що приймають різні типи вхідних параметрів.

Реалізація першого етапу має базову сигнатуру:

```
function (m::hNODEVocoder)(sound::AbstractMatrix, ps, st)
    end
```

Вона очікує на вхід лише партії звукових сигналів, які будуть проходити повну обробку. Цей етап обчислює значення початкових точок для інтегрування в латентному просторі та передає їх наступній функції.

Реалізація другого етапу має наступну сигнатуру:

```
function (m::hNODEVocoder)(
    (sound, (u0,)), ps, st
) end
```

Вона приймає на вхід партії звукових сигналів, а також початкові точки для інтегрування вкладеної NODE в латентному просторі. Цей крок вилучає характеристики вхідних сигналів та передає їх, разом зі значеннями u_0, останньому етапу для безпосереднього обчислення траєкторій.

Третій етап реалізується функцією з наступною сигнатурою:

```
function (m::hNODEVocoder)(
    (_, (u0, features)), ps, st
) end
```

Саме ця функція буде використовуватися під час навчання моделі, оскільки вона очікує характеристики звуку у якості одного з параметрів та початкової точки в латентному просторі для обчислення траєкторії.

Підготовлені тренувальні дані мають наступну форму:

```
NamedTuple{
    input::Matrix{Float32},
    target::Matrix{Float32},
    features::NamedTuple{
        f0s::Matrix{Float32},
        loudness::Matrix{Float32},
        mfccs::Matrix{Float32}
    }
}
```

Як видно, дані являють собою іменований кортеж, що містить три компоненти:

- `input` – партії відфільтрованих звукових сигналів з домішками шуму, що будуть подаватися на вхід моделі-кодувальнику.
- `target` – партії відфільтрованих звукових сигналів без домішок шуму, що будуть використовуватися в базовому критерії функції втрат для порівняння відхилення бажаної траєкторії від згенерованої.
- `features` – партії характеристик звукових сигналів, подані у формі матриць.

Для безпосереднього тренування моделі використаємо інструменти, реалізовані у Lux.jl, а також оптимізатор ADAM з пакета Optimisers.jl [119], [120], бібліотеку для диференційовного програмування Zygote.jl для автоматичного обчислення градієнтів функцій, та засоби з бібліотеки MLUtils.jl [121] для організації тренувальних даних.

3.6 Перевірка роботи моделі на тестовому сигналі

Для перевірки роботи моделі використаємо запис `103-1241-0019.flac` з датасета LibriSpeech, який не приймав участі у її тренуванні. Порівняння оригінального запису та його реконструкції з вилучених характеристик за допомогою розробленої моделі наведено на Рис. 3.11. Як видно на рисунку, обидва сигнали виглядають подібно, але про ступінь їх подібності важко судити через велику кількість точок на графіку. Для більш наочного порівняння також наведемо спектрограми сигналів у мел-шкалі, що наведені на Рис. 3.12.

Розглянемо фрагмент сигналу довжиною 512 семплів, що починається з 793.125 мілісекунди запису. Перенесемо кожне значення із фрагмента у латентний простір, застосувавши кодувальник нової моделі. Отримані латентні траєкторії наведені на Рис. 3.13. До цих траєкторій застосуємо декодувальник та порівняємо отриманий результат з оригінальним фрагментом. Таке порівняння неведено на Рис. 3.14. Як видно з рисунків, кодувальник породжує траєкторії в латентному просторі, що описують поведінку системи 3.20 з деякими параметрами, які можна отримати багатошаровим персептроном з визначних

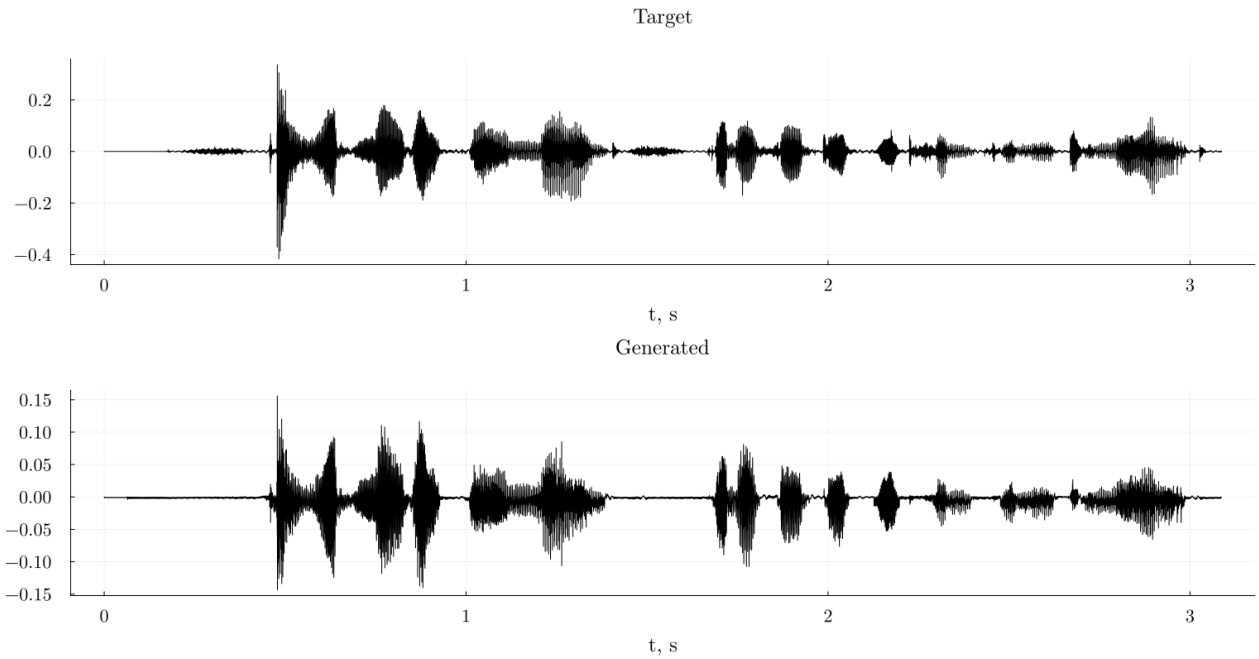


Рис. 3.11: Порівняння оригінального запису голосу та його реконструкції за допомогою нової моделі для параметричного синтезу голосу на основі нейромережевої архітектури hNODE.

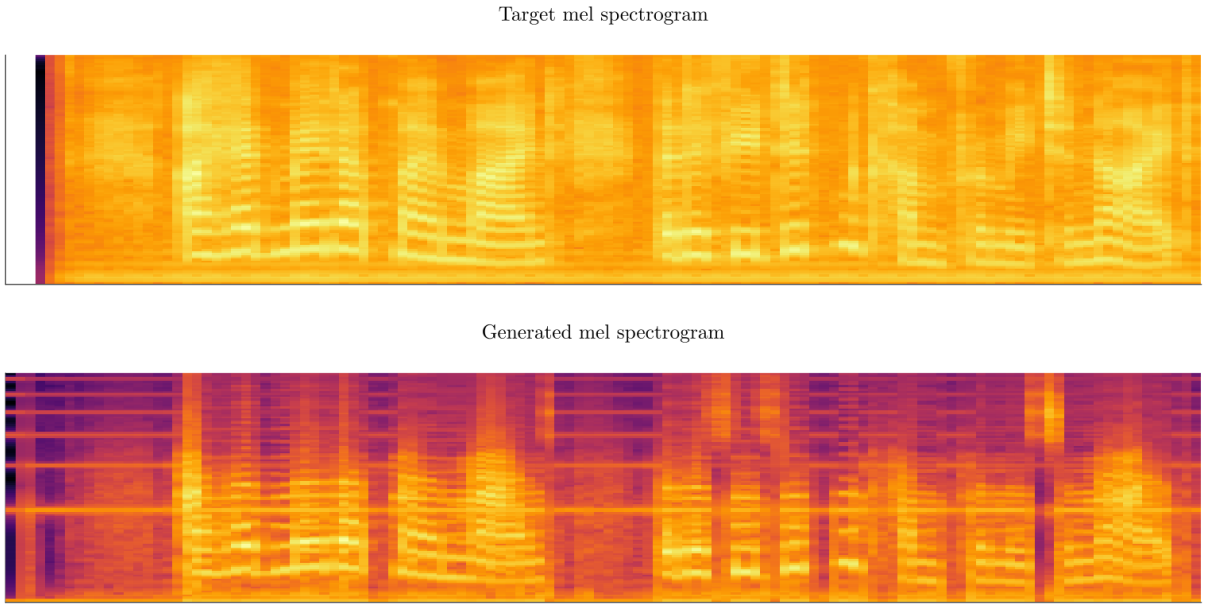


Рис. 3.12: Порівняння мел-спектрограм оригінального запису голосу та його реконструкції за допомогою нової моделі для параметричного синтезу голосу на основі нейромережевої архітектури hNODE.

характеристик поточного звукового сигналу. Якщо замість траєкторій, отриманих кодувальником, використати траєкторії, згенеровані шляхом інтегрування вкладеної NODE, отримаємо реконструкцію оригінального сигналу. Порівняння оригіналу та реконструкції наведено на Рис. 3.15.

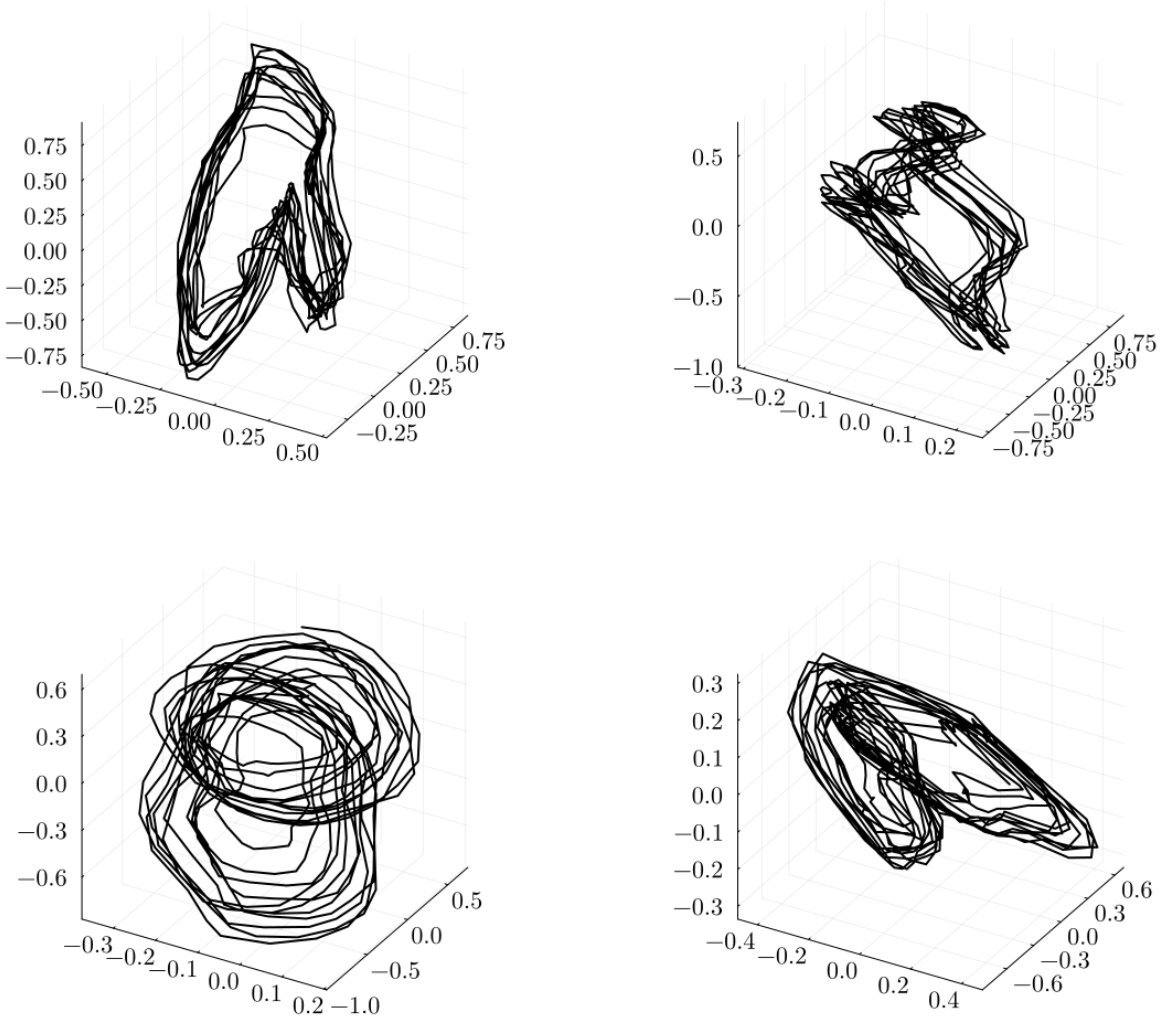


Рис. 3.13: Проєкції латентних траєкторій звукового сигналу, отриманих за допомогою кодувальника, на різні комбінації вісей десятивимірного простору.

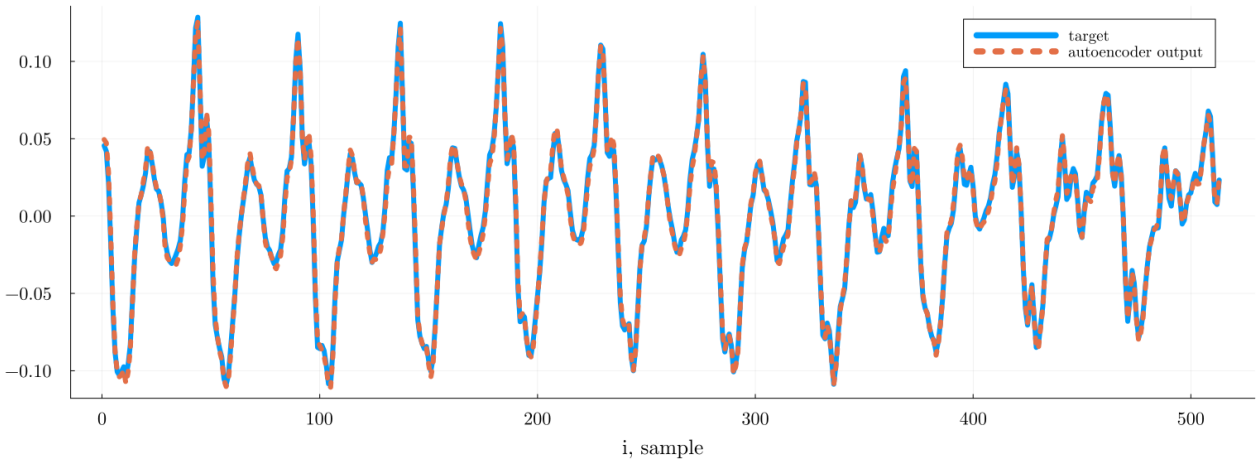


Рис. 3.14: Перетворення траєкторій в латентрому просторі у фрагмент звукового сигналу за допомогою декодувальника.

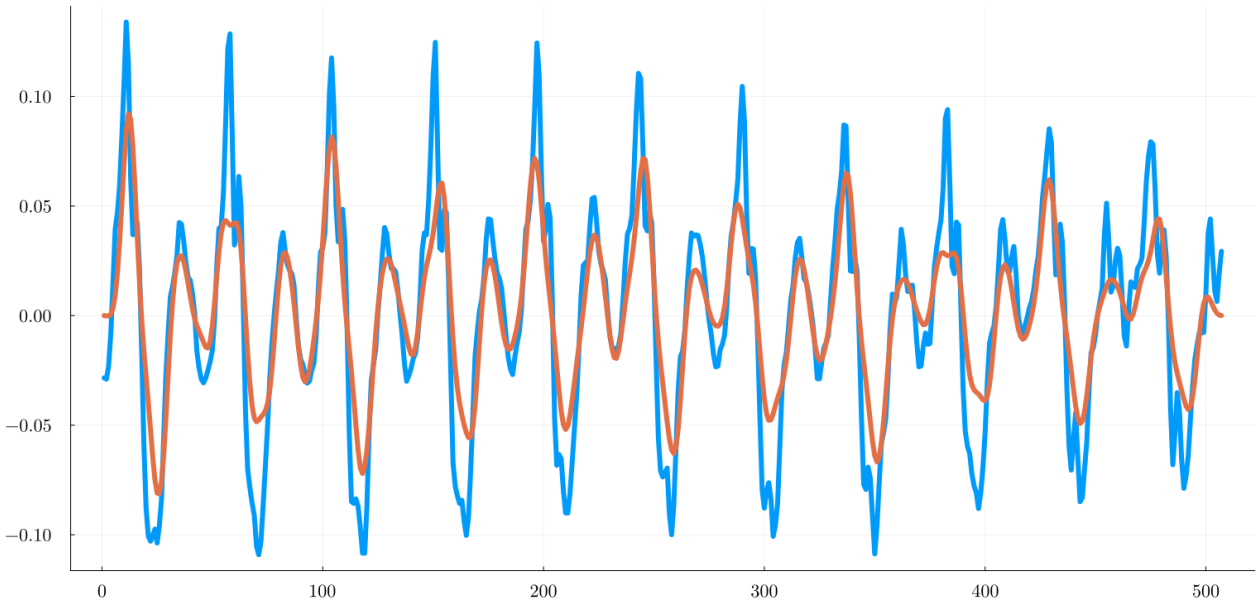


Рис. 3.15: Реконструкція фрагмента мовлення з визначних характеристик голосу за допомогою моделі hNODE.

3.7 Висновки

У розділі описано реалізацію нової моделі для аналізу та синтезу мовлення на основі нейромережевої архітектури hNODE. Запропоновано та обґрунтовано конкретну структуру кожного з компонентів моделі, а також проведено аналіз і перевірено коректність методів для вилучення характеристик голосового сигналу. Вичерпно описано принцип роботи моделі, та спосіб її використання й навчання.

Нову модель можна використовувати не тільки для подальших досліджень, але і для розробок практичних застосунків, наприклад, програм для потокового перетворення голосу.

Загальні висновки

У дисертаційній роботі були розглянуті методи для аналізу та синтезу мовлення, класичні та нейромережеві підходи до аналізу часових рядів, створено та досліджено властивості нової нейромережевої архітектури для моделювання часових рядів, а також поставлено і розв'язано задачу для моделювання мовлення як одновимірного часового ряду за допомогою розробленої нейромережевої моделі.

Результатами роботи є:

- Новий тип нейромережевої архітектури – hNODE, що дозволяє моделювати нестаціонарні часові ряди шляхом отримання керування для системи звичайних диференціальних рівнянь та побудови бієкції між сегментами часового ряду та точками отриманих траєкторій системи.

- Створення на основі розробленої архітектури нової моделі для синтезу мовлення, що здатна працювати за будь-яких значень частоти дискретизації без втрати попереднього навчання. В рамках нової моделі також реалізовано покращену версію алгоритму DIO для оцінки фундаментальної частоти звукового сигналу; програмне забезпечення, що реалізує покращений алгоритм, написано мовою Julia з дотриманням найкращих практик, що дозволяє його використання в подальших розробках та дослідженнях. Також реалізовано версію алгоритму CheapTrick для оцінки спектральної оболонки, що здатна працювати у режимі покадрової обробки звукових сигналів в реальному часі. Програмне забезпечення, що реалізує алгоритм, також було написано мовою Julia.

- Сформульовані й доведені теореми, а також проведені чисельні експерименти, що обґрунтовують і демонструють доцільність вибору архітектури та функції активації для нової моделі.

- Практичний порівняльний аналіз класичних та нейромережевих методів вкладення даних одновимірного часового ряду для реконструкції атракторів динамічних систем та їх застосування для отримання латентних траєкторій для нестаціонарних часових рядів.

Список використаних джерел

[1] D. Stanton, Y. Wang, and R. Skerry-Ryan, «Predicting expressive speaking style from text in end-to-end speech synthesis», in *2018 IEEE Spoken Language Technology Workshop (SLT)*, IEEE, 2018, pp. 595–602.

[2] Y. Ren, Y. Ruan, X. Tan, *et al.*, «Fastspeech: Fast, robust and controllable text to speech», *Advances in neural information processing systems*, vol. 32, 2019.

[3] G. Sun, Y. Zhang, R. J. Weiss, *et al.*, «Generating diverse and natural text-to-speech samples using a quantized fine-grained vae and autoregressive prosody prior», in *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, IEEE, 2020, pp. 6699–6703.

[4] Z. Zeng, J. Wang, N. Cheng, and J. Xiao, «Prosody learning mechanism for speech synthesis system without text length limit», *arXiv:2008.05656*, 2020.

[5] Y. Ren, C. Hu, X. Tan, *et al.*, «Fastspeech 2: Fast and high-quality end-to-end text to speech», *arXiv preprint arXiv:2006.04558*, 2020.

[6] M. Chen, X. Tan, B. Li, *et al.*, «Adaspeech: Adaptive text to speech for custom voice», *arXiv preprint arXiv:2103.00993*, 2021.

[7] S. Q. Abd Al-Rahman and S. A. Jassim, «Using deep learning neural networks to recognize and authenticate the identity of the speaker», in *2024 21st International Multi-Conference on Systems, Signals & Devices (SSD)*, IEEE, 2024, pp. 185–192.

[8] L. Wang, Z. Li, and K. Chen, «The text-dependent chinese speaker recognition system based on the universal individual identification», in *2021 IEEE 9th International Conference on Information, Communication and Networks (ICICN)*, IEEE, 2021, pp. 58–64.

[9] N. N. An, N. Q. Thanh, and Y. Liu, «Deep cnns with self-attention for speaker identification», *IEEE access*, vol. 7, pp. 85 327–85 337, 2019.

[10] R. Chakroun and M. Frikha, «Robust text-independent speaker recognition with short utterances using gaussian mixture models», in *2020 International Wireless Communications and Mobile Computing (IWCMC)*, IEEE, 2020, pp. 2204–2209.

[11] R. Chakroun and M. Frikha, «An improved system for large population text independent speaker recognition with short utterances», in *2021 International Wireless Communications and Mobile Computing (IWCMC)*, IEEE, 2021, pp. 2127–2131.

[12] M. M. Kabir, M. F. Mridha, J. Shin, I. Jahan, and A. Q. Ohi, «A survey of speaker recognition: Fundamental theories, recognition methods and opportunities», *IEEE Access*, vol. 9, pp. 79 236–79 263, 2021.

[13] T. En-Najjary, O. Rosec, and T. Chonavel, «A voice conversion method based on joint pitch and spectral envelope transformation.», in *INTERSPEECH*, 2004, pp. 1225–1228.

[14] H. Duxans, A. Bonafonte, A. Kain, and J. Van Santen, «Including dynamic and phonetic information in voice conversion systems», in *Proc. ICSLP*, vol. 4, 2004, pp. 5–8.

[15] H. Doi, T. Toda, T. Nakano, M. Goto, and S. Nakamura, «Singing voice conversion method based on many-to-many eigenvoice conversion and training data generation using a singing-to-singing synthesis system», in *Proceedings of The 2012 Asia Pacific Signal and Information Processing Association Annual Summit and Conference*, IEEE, 2012, pp. 1–6.

[16] S. Furui, T. Kikuchi, Y. Shinnaka, and C. Hori, «Speech-to-text and speech-to-speech summarization of spontaneous speech», *IEEE Transactions on Speech and Audio Processing*, vol. 12, no. 4, pp. 401–408, 2004.

[17] S. Bansal, H. Kamper, A. Lopez, and S. Goldwater, «Towards speech-to-text translation without speech recognition», *arXiv preprint arXiv:1702.03856*, 2017.

[18] A. Bérard, O. Pietquin, C. Servan, and L. Besacier, «Listen and translate: A proof of concept for end-to-end speech-to-text translation», *arXiv preprint arXiv:1612.01744*, 2016.

[19] C. Cieri, D. Miller, and K. Walker, «The fisher corpus: A resource for the next generations of speech-to-text.», in *LREC*, vol. 4, 2004, pp. 69–71.

[20] J. Bourquin, H. Schmidli, P. van Hoogevest, and H. Leuenberger, «Comparison of artificial neural networks (ann) with classical modelling techniques using different experimental designs and data from a galenical study on a solid dosage form», *European journal of pharmaceutical sciences*, vol. 6, no. 4, pp. 287–300, 1998.

[21] Y. Cho, F. Khosravikia, and E. M. Rathje, «A comparison of artificial neural network and classical regression models for earthquake-induced slope displacements», *Soil Dynamics and Earthquake Engineering*, vol. 152, p. 107 024, 2022.

[22] M. E. Nor, H. M. Safuan, N. F. M. Shab, *et al.*, «Neural network versus classical time series forecasting models», in *AIP Conference Proceedings*, AIP Publishing, vol. 1842, 2017.

[23] Z. Lu, H. Pu, F. Wang, Z. Hu, and L. Wang, «The expressive power of neural networks: A view from the width», *Advances in neural information processing systems*, vol. 30, 2017.

[24] I. Safran and O. Shamir, «Depth-width tradeoffs in approximating natural functions with neural networks», in *International conference on machine learning*, PMLR, 2017, pp. 2979–2987.

[25] M. Z. Alom, T. Josue, M. N. Rahman, W. Mitchell, C. Yakopcic, and T. M. Taha, «Deep versus wide convolutional neural networks for object recognition on neuromorphic system», in *2018 International Joint Conference on Neural Networks (IJCNN)*, IEEE, 2018, pp. 1–8.

[26] V. Y. Belozyorov, D. V. Dantsev, and Y. V. Koshel, «On the equivalence of real dynamic process and its neural network quadratic models», *Journal of Optimization, Differential Equations and Their Applications*, vol. 29, no. 2, pp. 76–91, 2021.

[27] Y. V. Koshel and V. Y. Belozyorov, «Univariate time series analysis with hyper neural ode», *Journal of Optimization, Differential Equations and Their Applications*, vol. 31, no. 2, pp. 50–66, 2023.

[28] V. Y. Belozyorov and Y. V. Koshel, «On systems of neural odes with generalized power activation functions», *Journal of Optimization, Differential Equations and Their Applications*, vol. 32, no. 2, pp. 56–91, 2024.

[29] M. W. Hirsch, «The dynamical systems approach to differential equations», *Bulletin of the American mathematical society*, vol. 11, no. 1, pp. 1–64, 1984.

[30] E. P. Wigner, «The unreasonable effectiveness of mathematics in the natural sciences», in *Mathematics and science*, World Scientific, 1990, pp. 291–306.

[31] O. M. Sharkovsky, «Dynamical systems theory», in *Encyclopedia of Modern Ukraine*, NASU Institute of Encyclopaedic Research, 2007. [Online]. Available: `https://esu.com.ua/article-24339`.

[32] V. V. Nemytskii and V. V. Stepanov, *Qualitative theory of differential equations*. OGIZ. Union of the State Book and Magazine Publishers, 1947.

[33] I. G. Koshevnikova, «Hausdorff dimension», in *Encyclopedia of Mathematics*, 2014. [Online]. Available: `http://encyclopediaofmath.org/index.php?title=Hausdorff_dimension&oldid=35823`.

[34] N. H. Packard, J. P. Crutchfield, J. D. Farmer, and R. S. Shaw, «Geometry from a time series», *Physical review letters*, vol. 45, no. 9, p. 712, 1980.

[35] F. Takens, «Detecting strange attractors in turbulence», in *Dynamical Systems and Turbulence, Warwick 1980: proceedings of a symposium held at the University of Warwick 1979/80*, Springer, 2006, pp. 366–381.

[36] L. Ljung, «System identification», in *Encyclopedia of Mathematics*, 2023. [Online]. Available: `http://encyclopediaofmath.org/index.php`.

[37] C. M. Legaard, T. Schranz, G. Schweiger, *et al.*, «Constructing neural network-based models for simulating dynamical systems», *CoRR*, vol. abs/2111.01495, 2021. arXiv: `2111.01495`. [Online]. Available: `https://arxiv.org/abs/2111.01495`.

[38] L. Vu-Quoc, *File:neuron3.png — wikimedia commons, the free media repository*, [Online; accessed 14-January-2025], 2023. [Online]. Available: https://commons.wikimedia.org/w/index.php?title=File:Neuron3.png&oldid=757974670.

[39] Y. LeCun, B. Boser, J. Denker, *et al.*, «Handwritten digit recognition with a back-propagation network», in *Advances in Neural Information Processing Systems*, D. Touretzky, Ed., vol. 2, Morgan-Kaufmann, 1989, pp. 396–404.

[40] A. Baldominos, Y. Saez, and P. Isasi, «A survey of handwritten character recognition with mnist and emnist», *Applied Sciences*, vol. 9, no. 15, 2019, ISSN: 2076-3417. DOI: 10.3390/app9153169. [Online]. Available: https://www.mdpi.com/2076-3417/9/15/3169.

[41] K. Hornik, «Approximation capabilities of multilayer feedforward networks», *Neural Networks*, vol. 4, no. 2, pp. 251–257, 1991, ISSN: 0893-6080. DOI: https://doi.org/10.1016/0893-6080(91)90009-T.

[42] L. S. Blackford, A. Petitet, R. Pozo, *et al.*, «An updated set of basic linear algebra subprograms (blas)», *ACM Transactions on Mathematical Software*, vol. 28, no. 2, pp. 135–151, 2002.

[43] R. T. Chen, Y. Rubanova, J. Bettencourt, and D. K. Duvenaud, «Neural ordinary differential equations», *Advances in neural information processing systems*, vol. 31, 2018.

[44] A. Ambardar *et al.*, *Analog and digital signal processing*. PWS Boston, MA, USA, 1995.

[45] T. Halmrast, «Cepstrum; a "forgotten" analysis?"», in *BNAM, Baltic-Nordic Acoustics Meeting*, 2021. [Online]. Available: https://www.norskakustikselskap.org/wp-content/uploads/2021/10/Halmrast_Cepstrum_BNAM2021_01-11-14-Halmrast-Tor.pdf.

[46] S. Furui, «Cepstral analysis technique for automatic speaker verification», *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 29, no. 2, pp. 254–272, 1981.

[47] F.-s. Wei and M. Li, «Cepstrum analysis of seismic source characteristics», *Acta Seismologica Sinica*, vol. 16, pp. 50–58, 2003.

[48] R. B. Randall, «A history of cepstrum analysis and its application to mechanical problems», *Mechanical Systems and Signal Processing*, vol. 97, pp. 3–19, 2017.

[49] R. Carlson, «Models of speech synthesis», *Proceedings of the National Academy of Sciences*, vol. 92, no. 22, pp. 9932–9937, 1995.

[50] M. Liberman, «Linguistics 520: Lecture notes 2», 2013. [Online]. Available: `https://www.ling.upenn.edu/courses/Fall_2013/ling520/LectureNotes2.html`.

[51] M. R. Lvov, *Fundamentals of speech theory: A textbook for students of higher educational institutions*, 2000.

[52] A. Norcliffe, C. Bodnar, B. Day, J. Moss, and P. Liò, «Neural ode processes», *arXiv preprint arXiv:2103.12413*, 2021.

[53] M. Chalvidal, M. Ricci, R. VanRullen, and T. Serre, «Go with the flow: Adaptive control for neural odes», *arXiv preprint arXiv:2006.09545*, 2020.

[54] V. Y. Belozyorov and D. V. Dantsev, «Stability of neural ordinary differential equations with power nonlinearities», *Journal of Optimization, Differential Equations and Their Applications*, vol. 28, no. 2, pp. 21–46, 2020.

[55] V. Belozyorov and D. Dantsev, «Modeling of chaotic processes by means of antisymmetric neural odes», *Journal of Optimization, Differential Equations and Their Applications*, vol. 30, no. 1, pp. 1–41, 2022. DOI: `10.21203/rs.3.rs-790408`.

[56] L. M. Pecora, L. Moniz, J. Nichols, and T. L. Carroll, «A unified approach to attractor reconstruction», *Chaos: An Interdisciplinary Journal of Nonlinear Science*, vol. 17, no. 1, 2007.

[57] B. Chang, M. Chen, E. Haber, and E. H. Chi, «Antisymmetricrnn: A dynamical system view on recurrent neural networks», *arXiv e-prints*, arXiv–1902.09689, 2019.

[58] A. R. Barron, «Universal approximation bounds for superpositions of a sigmoidal function», *IEEE Transactions on Information theory*, vol. 39, no. 3, pp. 930–945, 1993.

[59] C. K. Chui, X. Li, and H. N. Mhaskar, «Limitations of the approximation capabilities of neural networks with one hidden layer», *Advances in Computational Mathematics*, vol. 5, no. 1, pp. 233–243, 1996.

[60] D. Elbrächter, D. Perekrestenko, P. Grohs, and H. Bölcskei, «Deep neural network approximation theory», *IEEE Transactions on Information Theory*, vol. 67, no. 5, pp. 2581–2623, 2021.

[61] K. Hornik, M. Stinchcombe, and H. White, «Multilayer feedforward networks are universal approximators», *Neural networks*, vol. 2, no. 5, pp. 359–366, 1989.

[62] J. Wu, «Introduction to convolutional neural networks», *National Key Lab for Novel Software Technology. Nanjing University. China*, vol. 5, no. 23, p. 495, 2017.

[63] Z. Li, F. Liu, W. Yang, S. Peng, and J. Zhou, «A survey of convolutional neural networks: Analysis, applications, and prospects», *IEEE transactions on neural networks and learning systems*, vol. 33, no. 12, pp. 6999–7019, 2021.

[64] K. He, X. Zhang, S. Ren, and J. Sun, «Deep residual learning for image recognition», in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[65] M. S. Hanif and M. Bilal, «Competitive residual neural network for image classification», *ICT Express*, vol. 6, no. 1, pp. 28–37, 2020.

[66] A. C. Tsoi and A. Back, «Discrete time recurrent neural network architectures: A unifying review», *Neurocomputing*, vol. 15, no. 3-4, pp. 183–223, 1997.

[67] L. R. Medsker, L. Jain, *et al.*, «Recurrent neural networks», *Design and Applications*, vol. 5, no. 64-67, p. 2, 2001.

[68] S. Hochreiter, «Long short-term memory», *Neural Computation MIT-Press*, 1997.

[69] Y. Yu, X. Si, C. Hu, and J. Zhang, «A review of recurrent neural networks: Lstm cells and network architectures», *Neural computation*, vol. 31, no. 7, pp. 1235–1270, 2019.

[70] A. Vaswani, «Attention is all you need», *Advances in Neural Information Processing Systems*, 2017.

[71] K. T. Chitty-Venkata, M. Emani, V. Vishwanath, and A. K. Somani, «Neural architecture search for transformers: A survey», *IEEE Access*, vol. 10, pp. 108 374–108 412, 2022.

[72] M. Sander, P. Ablin, and G. Peyré, «Do residual neural networks discretize neural ordinary differential equations?», *Advances in Neural Information Processing Systems*, vol. 35, pp. 36 520–36 532, 2022.

[73] A. Gu and T. Dao, «Mamba: Linear-time sequence modeling with selective state spaces», *arXiv preprint arXiv:2312.00752*, 2023.

[74] S. Cuomo, V. S. Di Cola, F. Giampaolo, G. Rozza, M. Raissi, and F. Piccialli, «Scientific machine learning through physics–informed neural networks: Where we are and what's next», *Journal of Scientific Computing*, vol. 92, no. 3, p. 88, 2022.

[75] M. Raissi, P. Perdikaris, and G. E. Karniadakis, «Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations», *Journal of Computational physics*, vol. 378, pp. 686–707, 2019.

[76] Z. Xiang, W. Peng, X. Liu, and W. Yao, «Self-adaptive loss balanced physics-informed neural networks», *Neurocomputing*, vol. 496, pp. 11–34, 2022.

[77] K. Khalil, *Nonlinear Systems, 2nd Edition*. Englewood Cliffs, NJ, Prentice-Hall, 1996.

[78] V. Y. Belozyorov and S. A. Volkova, «Role of logistic and ricker's maps in appearance of chaos in autonomous quadratic dynamical systems», *Nonlinear Dynamics*, vol. 83, pp. 719–729, 2016.

[79] V. Ye Belozyorov, «On novel conditions of chaotic attractors existence in autonomous polynomial dynamical systems», *Nonlinear Dynamics*, vol. 91, no. 4, pp. 2435–2452, 2018.

[80] V. Y. Belozyorov, «Universal approach to the problem of emergence of chaos in autonomous dynamical systems», *Nonlinear Dynamics*, vol. 95, no. 1, pp. 579–595, 2019.

[81] V. Y. Belozyorov and S. A. Volkova, «Discrete processes and chaos in systems of ordinary differential equations», *Journal of Optimization, Differential Equations and Their Applications*, vol. 30, no. 2, pp. 62–78, 2022.

[82] M. J. Feigenbaum, «Quantitative universality for a class of nonlinear transformations», *Journal of statistical physics*, vol. 19, no. 1, pp. 25–52, 1978.

[83] T. Tsuchiya and D. Yamagishi, «The complete bifurcation diagram for the logistic map», *Zeitschrift für Naturforschung A*, vol. 52, no. 6-7, pp. 513–516, 1997.

[84] V. Panayotov, G. Chen, D. Povey, and S. Khudanpur, «Librispeech: An asr corpus based on public domain audio books», in *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2015, pp. 5206–5210. DOI: 10.1109/ICASSP.2015.7178964.

[85] G. Datseris and U. Parlitz, *Nonlinear dynamics: a concise introduction interlaced with code*. Springer Nature, 2022.

[86] P. de la Cuadra, A. S. Master, and C. S. Sapp, «Efficient pitch detection techniques for interactive music», in *International Conference on Mathematics and Computing*, 2001. [Online]. Available: https://api.semanticscholar.org/CorpusID:18771948.

[87] D. Jouvet and Y. Laprie, «Performance analysis of several pitch detection algorithms on simulated and real noisy speech data», in *2017 25th european signal processing conference (eusipco)*, IEEE, 2017, pp. 1614–1618.

[88] M. Morise, F. Yokomori, and K. Ozawa, «World: A vocoder-based high-quality speech synthesis system for real-time applications», *IEICE TRANSACTIONS on Information and Systems*, vol. 99, no. 7, pp. 1877–1884, 2016.

[89] R. Daido and Y. Hisaminato, «A fast and accurate fundamental frequency estimator using recursive moving average filters.», in *INTERSPEECH*, 2016, pp. 2160–2164.

[90] S. Kornblith, G. Lynch, M. Holters, *et al.*, *Juliadsp/dsp.jl: V0.7.9*, Sep. 2023. DOI: 10.5281/zenodo.8344531. [Online]. Available: https://doi.org/10.5281/zenodo.8344531.

[91] D. Schwarz, «Spectral envelopes in sound analysis and synthesis», Ph.D. dissertation, Universität Stuttgart, Fakultät Informatik, 1998.

[92] N. Dave, «Feature extraction methods lpc, plp and mfcc in speech recognition», *International journal for advance research in engineering and technology*, vol. 1, no. 6, pp. 1–4, 2013.

[93] J. Barros and R. I. Diego, «On the use of the hanning window for harmonic analysis in the standard framework», *IEEE transactions on power delivery*, vol. 21, no. 1, pp. 538–539, 2005.

[94] V. Gupta and M. Mittal, «Qrs complex detection using stft, chaos analysis, and pca in standard and real-time ecg databases», *Journal of The Institution of Engineers (India): Series B*, vol. 100, no. 5, pp. 489–497, 2019.

[95] M. K. Kıymık, İ. Güler, A. Dizibüyük, and M. Akın, «Comparison of stft and wavelet transform methods in determining epileptic seizure activity in eeg signals for real-time application», *Computers in biology and medicine*, vol. 35, no. 7, pp. 603–616, 2005.

[96] Z. Tüske, P. Golik, R. Schlüter, and F. R. Drepper, «Non-stationary feature extraction for automatic speech recognition», in *2011 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, IEEE, 2011, pp. 5204–5207.

[97] B. Chakraborty, «A proposal for classification of multisensor time series data based on time delay embedding», *International Journal on Smart Sensing and Intelligent Systems*, vol. 7, no. 5, pp. 1–5, 2020.

[98] J. Frank, S. Mannor, and D. Precup, «Activity and gait recognition with time-delay embeddings», in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 24, 2010, pp. 1581–1586.

[99] C. Nalmpantis and D. Vrakas, «Signal2vec: Time series embedding representation», in *International conference on engineering applications of neural networks*, Springer, 2019, pp. 80–90.

[100] L. Cao, «Practical method for determining the minimum embedding dimension of a scalar time series», *Physica D: Nonlinear Phenomena*, vol. 110, no. 1-2, pp. 43–50, 1997.

[101] R. Hegger and H. Kantz, «Improved false nearest neighbor method to detect determinism in time series data», *Physical Review E*, vol. 60, no. 4, p. 4970, 1999.

[102] M. B. Kennel, R. Brown, and H. D. Abarbanel, «Determining embedding dimension for phase-space reconstruction using a geometrical construction», *Physical review A*, vol. 45, no. 6, p. 3403, 1992.

[103] A. Krakovská, K. Mezeiová, and H. Budáčová, «Use of false nearest neighbours for selecting variables and embedding parameters for state space reconstruction», *Journal of Complex Systems*, vol. 2015, no. 1, p. 932 750, 2015.

[104] D. S. Broomhead and G. P. King, «Extracting qualitative dynamics from experimental data», *Physica D: Nonlinear Phenomena*, vol. 20, no. 2-3, pp. 217–236, 1986.

[105] G. Datseris, «Dynamicalsystems. jl: A julia software library for chaos and nonlinear dynamics», *Journal of Open Source Software*, vol. 3, no. 23, p. 598, 2018.

[106] K. Geist, U. Parlitz, and W. Lauterborn, «Comparison of different methods for computing lyapunov exponents», *Progress of theoretical physics*, vol. 83, no. 5, pp. 875–893, 1990.

[107] E. Koshel, «Local jacobian estimation for delay embedded time series data», in *XVII International scientific and practical conference, Dnipro*, 2020, pp. 152–153.

[108] A. Pal, *Lux: Explicit Parameterization of Deep Neural Networks in Julia*, version v0.5.0, Apr. 2023. DOI: 10.5281/zenodo.7808904. [Online]. Available: https://doi.org/10.5281/zenodo.7808904.

[109] N. Sato and Y. Obuchi, «Emotion recognition using mel-frequency cepstral coefficients», *Information and Media Technologies*, vol. 2, no. 3, pp. 835–848, 2007. DOI: 10.11185/imt.2.835.

[110] X. Zhou, D. Garcia-Romero, R. Duraiswami, C. Espy-Wilson, and S. Shamma, «Linear versus mel frequency cepstral coefficients for speaker recognition», in *2011 IEEE Workshop on Automatic Speech Recognition and Understanding*, 2011, pp. 559–564. DOI: 10.1109/ASRU.2011.6163888.

[111] R. Vergin, D. O'Shaughnessy, and A. Farhat, «Generalized mel frequency cepstral coefficients for large-vocabulary speaker-independent continuous-speech recognition», *IEEE Transactions on Speech and Audio Processing*, vol. 7, no. 5, pp. 525–532, 1999. DOI: 10.1109/89.784104.

[112] S. S. Stevens, J. Volkmann, and E. B. Newman, «A scale for the measurement of the psychological magnitude pitch», *The journal of the acoustical society of america*, vol. 8, no. 3, pp. 185–190, 1937.

[113] O. Douglas and O. Shaughnessy, «Speech communications: Human and machine», *IEEE press, Newyork*, pp. 367–433, 2000.

[114] X. Ying, «An overview of overfitting and its solutions», *Journal of Physics: Conference Series*, vol. 1168, no. 2, p. 022 022, Feb. 2019. DOI: 10.1088/1742-6596/1168/2/022022. [Online]. Available: https://dx.doi.org/10.1088/1742-6596/1168/2/022022.

[115] T. van Laarhoven, «L2 regularization versus batch and weight normalization», *CoRR*, vol. abs/1706.05350, 2017. arXiv: 1706.05350. [Online]. Available: http://arxiv.org/abs/1706.05350.

[116] R. Moore and J. DeNero, «L1 and l2 regularization for multiclass hinge loss models», in *Symposium on machine learning in speech and language processing*, 2011.

[117] G. Shi, J. Zhang, H. Li, and C. Wang, «Enhance the performance of deep neural networks via l2 regularization on the input of activations», *Neural Processing Letters*, vol. 50, no. 1, pp. 57–75, Aug. 2019, ISSN: 1573-773X. DOI: 10.1007/s11063-018-9883-8. [Online]. Available: https://doi.org/10.1007/s11063-018-9883-8.

[118] M. Innes, «Don't unroll adjoint: Differentiating ssa-form programs», *CoRR*, vol. abs/1810.07951, 2018. arXiv: 1810.07951. [Online]. Available: http://arxiv.org/abs/1810.07951.

[119] M. Innes, «Flux: Elegant machine learning with julia», *Journal of Open Source Software*, 2018. DOI: 10.21105/joss.00602.

[120] M. Innes, E. Saba, K. Fischer, *et al.*, «Fashionable modelling with flux», *CoRR*, vol. abs/1811.01457, 2018. arXiv: 1811.01457. [Online]. Available: https://arxiv.org/abs/1811.01457.

[121] C. Lucibello, *Mlutils.jl*, https://juliaml.github.io/MLUtils.jl/stable/, Feb. 2022.

Додаток А

СПИСОК ПУБЛІКАЦІЙ ЗА ТЕМОЮ ДИСЕРТАЦІЇ

Кошеля Євгена Васильовича

№ з/п	Назва	Характер роботи	Вихідні дані	Обсяг	Співавтори
1	On the Equivalence of Real Dynamic Process and Its Models	Стаття	Journal of Optimization, Differential Equations and Their Applications. DOI 10.15421/142109, ISSN (print) 2617–0108, ISSN (on-line) 2663–6824. - V. 29. - №2.- December 2021. - P. 76-91.	16 сторінок	Belozyorov V. Ye., Dantsev D. V.
2	Univariate time series analysis with hyperneural ODE	Стаття	Journal of Optimization, Differential Equations and Their Applications. DOI 10.15421/142311, ISSN (print) 2617–0108, ISSN (on-line) 2663–6824. - V. 31. - №2.- December 2023. - P. 50-66.	17 сторінок	Belozyorov V. Ye.
3	On systems of neural ODEs with generalized power activation functions	Стаття	Journal of Optimization, Differential Equations and Their Applications. DOI 10.15421/141901, ISSN (print) 2617–0108, ISSN (on-line) 2663–6824. - V. 32. - №2.- December 2024. - P. 56 - 91.	36 сторінок	Belozyorov V. Ye.

№ з/п	Назва	Характер роботи	Вихідні дані	Обсяг	Співавтори
4	The possibility of the robust chaos emergence in the Lorenz-like nonautonomous system	Стаття	Journal of Optimization, Differential Equations and Their Applications. DOI 10.15421/141902, ISSN (print) 2617–0108, ISSN (on-line) 2663–6824. - V. 27. - №2.- December 2019. - P. 44-59.	16 сторінок	Belozyorov V. Ye., Zaytsev V.G.
5	Adaptation of the world framework for frame-by-frame real-time speech analysis	Стаття	System technologies. DOI10.34185/1562-9945-5-148-2023-03, ISSN (print) 1562-9945, ISSN (on-line) 2707-7977. - V. 5 - №148 - 2023 - P. 21-36.	16 сторінок	
6	Neural network-assisted continuous embedding of univariate data streams for time series analysis	Стаття	System technologies. DOI 10.34185/1562-9945-2-151-2024-08, ISSN (print) 1562-9945, ISSN (on-line) 2707-7977. - V. 2 - № 151 - 2024 - P. 92-101.	10 сторінок	
7	Local Jacobian estimation for delay embedded time series data	Тези	18-та міжнародна науково-практична конференція “Математичне та програмне забезпечення інтелектуальних систем” (MPZIS-2020). - 18 - 20 листопада 2020 р. Дніпро, с. 152 - 153.	2 сторінки	
8	Modelling Dynamical Processes Described By Time Series Data With Neural Differential Equations	Тези	20-та міжнародна науково-практична конференція “Математичне та програмне забезпечення інтелектуальних систем” (MPZIS-2022). - 23-25 листопада 2022 р. Дніпро, с. 111-112.	2 сторінки	Belozyorov V. Ye.

№ з/п	Назва	Характер роботи	Вихідні дані	Обсяг	Співавтори
9	Lyapunov spectrum estimation for delay embedded time series	Тези	19-та міжнародна науково-практична конференція “Математичне та програмне забезпечення інтелектуальних систем” (MPZIS-2021). - 17 - 19 листопада 2021 р. Дніпро, с. 110-111.	2 сторінки	
10	Hyper neural ODE for univariate time series analysis	Тези	21-ша міжнародна науково-практична конференція “Математичне та програмне забезпечення інтелектуальних систем” (MPZIS-2023). - 22-24 листопада 2023, Дніпро, с. 21-22.	2 сторінки	
11	Generation of stable systems of ordinary differential equations for univariate time series modeling	Тези	XXVII міжнародна конференція з актоматичного керування «Автоматика 2024» - 20 - 22 листопада 2024 р., Дніпро, С. 36 - 37.	2 сторінки	Belozyorov V. Ye.
12	On use of power activation functions with a degree less than 1 in neural networks	Тези	22-га міжнародна науково-практична конференція «Математичне та програмне забезпечення інтелектуальних систем» (MPZIS-2024). - 20–22 листопада 2024 р., Дніпро, С. 7 - 9.	3 сторінки	Belozyorov V. Ye.
13	Improvement for peaks and zero-crossings detection for time-domain pitch estimation algorithms	Тези	Proceedings of 1st international scientific and practical conference «CURRENT CHALLENGES OF SCIENCE AND EDUCATION», - September 18-20 2023, Berlin, - P. 116-118.	3 сторінки	