

**Міністерство освіти і науки України  
Національний технічний університет  
«Дніпровська політехніка»**

**Інститут електроенергетики  
Факультет інформаційних технологій  
Кафедра безпеки інформації та телекомунікацій**

**ПОЯСНЮВАЛЬНА ЗАПИСКА  
кваліфікаційної роботи ступеню бакалавра**

студента Черкашина Семена Борисовича

академічної групи 172–16зск–1

спеціальності 172 «Телекомунікації і радіотехніка»

за освітньо-професійною програмою Телекомунікації і радіотехніка

на тему Розробка імпульсної характеристики Колмогорова-Вінера для неперервних фрактальних процесів в телекомунікаційних мережах

Керівники	Прізвище, ініціали	Оцінка за шкалою		Підпис
		рейтинговою	інституційною	
кваліфікаційної роботи	Корнієнко В.І.			
розділів:				
спеціальний	Горєв В.М.			
економічний	Романюк Н.М.			
<b>Рецензент</b>	Шедловський І.А.			
<b>Нормоконтролер</b>	Гусєв О.Ю.			

**Дніпро  
2019**

**ЗАТВЕРДЖЕНО:**  
завідувач кафедри  
безпеки інформації та телекомунікацій  
д.т.н., професор \_\_\_\_\_ Корнієнко В.І.  
«\_\_\_» \_\_\_\_\_ 2019 року

**ЗАВДАННЯ**  
**на кваліфікаційну роботу ступеня бакалавра**

студенту Черкашину С.Б. академічної групи 172–16зск–1

спеціальності 172 Телекомунікації та радіотехніка

за освітньо-професійною програмою 172 «Телекомунікації і радіотехніка»

на тему: Розробка імпульсної характеристики Колмогорова-Вінера для неперервних фрактальних процесів в телекомунікаційних мережах

затверджену наказом ректора НТУ «Дніпровська політехніка» від \_\_\_\_\_ № \_\_\_\_\_

<b>Розділ</b>	<b>Зміст</b>	<b>Термін виконання</b>
Стан питання. Постановка задачі	Огляд літератури за темою, обґрунтування поставленої задачі	11.02.2019 – 11.03.2019
Спеціальна частина	Наближене обчислення вагової функції фільтра Колмогорова-Вінера для фрактальних процесів з степеневою структурною функцією у вигляді розвинення за поліномами Чебишева. Порівняння результатів з роботами попередників.	11.03.2019 – 01.05.2019
Економічний розділ	Розрахувати трудомісткість та витрати на дослідження фільтру Колмогорова-Вінера	01.05.2019 – 01.06.2019

Завдання видано

В.І. Корнієнко

Дата видачі 11.02.2019

Дата подання до екзаменаційної комісії

Прийнято до виконання

С.Б. Черкашин

## РЕФЕРАТ

**Пояснювальна записка:** 63с., 19 рис., 1 табл., 5 додатків, 9 джерел.

**Об'єкт дослідження:** фільтр Колмогорова-Вінера для неперервного фрактального процесу зі степеневою кореляційною функцією.

**Предмет дослідження:** вагова функція даного фільтру.

**Мета роботи:** отримати вагову функцію відповідного фільтра у вигляді обірваного розвинення за поліномами Чебишева. Порівняти результати з відповідними результатами для іншої системи поліномів.

У першому розділі наведено загальні відомості про фільтр Колмогорова-Вінера та оглянуто роботи попередників. Зроблено висновок, що задача, яка поставлена у роботі, є актуальною для телекомунікацій.

В спеціальній частині досліджено відповідні розв'язки у вигляді обірваних розвинень за поліномами Чебишева першого роду. Дослідження проведено до наближення 18 поліномів включно. Результати порівняно з результатами робіт (Gorev, Gusev and Korniienko, Radio Electronics, Computer Science, Control, No.2, 2019, in press) та (Gorev, Gusev and Korniienko, CEUR Workshop Proceedings, vol. 2353, 2019), де така задача досліджувалась на основі інших систем поліномів. Зроблено висновок, що поведінка наближень для різних систем поліномів є майже однаковою.

В економічному розділі розрахована трудомісткість та витрати на дослідження фільтру Колмогорова-Вінера.

Результати роботи можуть бути застосовані для прогнозування трафіку у телекомунікаційних системах, зокрема таких, як мережі стандарту IEEE 802.11b та для маршрутизаторів між внутрішніми мережами підприємств або університетів та мережею Інтернет.

ІНТЕГРАЛЬНЕ РІВНЯННЯ ФРЕДГОЛЬМА ПЕРШОГО РОДУ, ВАГОВА ФУНКЦІЯ ФІЛЬТРУ КОЛМОГОРОВА-ВІНЕРА, СТЕПЕНЕВА СТРУКТУРНА ФУНКЦІЯ, ПОЛІНОМИ ЧЕБИШЕВА ПЕРШОГО РОДУ

## РЕФЕРАТ

**Пояснительная записка:** 63с., 19 рис., 1 табл., 5 приложений, 9 источников.

**Объект исследования:** фильтр Колмогорова-Винера для непрерывного фрактального процесса со степенной корреляционной функцией.

**Предмет исследования:** весовая функция данного фильтра.

**Цель работы:** получить весовую функцию соответствующего фильтра в виде оборванного разложения по полиномам Чебышева. Сравнить результаты с соответствующими результатами для другой системы полиномов.

В первом разделе приведены общие сведения про фильтр Колмогорова–Винера и рассмотрены работы предшественников. Сделан вывод о том, что задача, поставленная в работе, является актуальной для телекоммуникаций.

В специальной части исследованы соответствующие решения в виде оборванных разложений по полиномам Чебышева. Исследование проведено до приближения 18 полиномов включительно. Результаты сравнены с работами (Gorev, Gusev and Korniienko, Radio Electronics, Computer Science, Control, No.2, 2019, in press) и (Gorev, Gusev and Korniienko, CEUR Workshop Proceedings, vol. 2353, 2019), где такая задача исследовалась на основе других систем полиномов. Сделан вывод о том, что поведение приближений для разных систем полиномов есть почти одинаковым.

В экономическом разделе рассчитана трудоемкость и затраты на исследование фильтра Колмогорова–Винера.

Результаты работы могут быть применены к прогнозированию трафика в телекоммуникационных системах, например таких, как сети стандарта IEEE 802.11b, и маршрутизаторов между внутренними сетями предприятий или университетов и сетью Интернет.

ИНТЕГРАЛЬНОЕ УРАВНЕНИЕ ФРЕДГОЛЬМА ПЕРВОГО РОДА,  
ВЕСОВАЯ ФУНКЦИЯ ФИЛЬТРА КОЛМОГОРОВА–ВИНЕРА, СТЕПЕННАЯ  
СТРУКТУРНАЯ ФУНКЦИЯ, ПОЛИНОМЫ ЧЕБЫШЕВА ПЕРВОГО РОДА

## ABSTRACT

**Explanatory note:** 63 pages, 19 pic., 1 tab., 5 appendices, 9 references.

**Object of research:** Kolmogorov-Wiener filter for a continuous fractal process with a power-law structure function.

**Subject of research:** the weight function of the filter.

**The aim of the work:** to obtain the weight function of the corresponding filter in the form of a truncated Chebyshev polynomial expansion and to compare the results with the corresponding results for another polynomial set.

In the first section the general information about the Kolmogorov–Wiener filter is given and a literature review is made. It is concluded that the problem under consideration is an urgent problem for telecommunications.

In the special part, the corresponding solutions in the form of the truncated Chebyshev polynomial expansion are investigated. The investigation is made up to the 18-polynomial approximation. The results are compared with the results of the papers (Gorev, Gusev and Korniienko, Radio Electronics, Computer Science, Control, No.2, 2019, in press) and (Gorev, Gusev and Korniienko, CEUR Workshop Proceedings, vol. 2353, 2019), where the problem under consideration is investigated on the basis of another polynomial sets. It is concluded that the behavior of the approximations for different polynomial sets is almost identical.

In the economic Section the labor intensity and the costs for the Kolmogorov–Wiener filter investigation are calculated.

The results of the work may be applied to the traffic forecast in telecommunication systems, for example, for the IEEE 802.11b networks and for the routers between the internal networks and the Internet.

FREDHOLM INTEGRAL EQUATION OF THE FIRST KIND,  
KOLMOGOROV–WIENER FILTER WEIGHT FUNCTION, POWER-LAW  
STRUCTURE FUNCTION, CHEBYSHEV POLYNOMIALS OF THE FIRST  
KIND.

## ЗМІСТ

ВСТУП.....	7
1 СТАН ПИТАННЯ І ПОСТАНОВКА ЗАДАЧІ.....	9
1.1 Стан питання.....	9
1.1.1 Загальні відомості про фільтр Колмогорова–Вінера.....	9
1.1.2 Огляд робіт попередників.....	11
1.2 Постановка задачі.....	15
1.3 Висновки.....	15
2 СПЕЦІАЛЬНА ЧАСТИНА.....	16
2.1 Система поліномів Чебишева, ортогональних на відрізку $t \in [0, T]$ .....	16
2.2 Обчислення вагової функції фільтра.....	18
2.3 Порівняння результатів з результатами для інших систем поліномів....	35
2.4 Висновки.....	36
3 ЕКОНОМІЧНИЙ РОЗДІЛ.....	38
3.1 Розрахунок трудомісткості дослідження фільтру Колмогорова–Вінера..	38
3.2 Розрахунок витрат на створення програми.....	41
3.3 Висновки.....	45
ВИСНОВКИ.....	46
ПЕРЕЛІК ПОСИЛАНЬ.....	49
ДОДАТОК А. Відомість матеріалів кваліфікаційної роботи.....	50
ДОДАТОК Б. Перелік документів на оптичному носії.....	51
ДОДАТОК В. Код у Wolfram Mathematica.....	52
В. 1 Розрахунок матриці інтегральних дужок та стовпчику з коефіцієнтів $b_m$ .....	52
В. 2 Розрахунок коефіцієнтів при поліномах та побудова відповідних графіків для наближень від одного до вісімнадцяти поліномів.....	52
ДОДАТОК Г. Відгук керівника економічного розділу.....	62
ДОДАТОК Д. Відгук керівника кваліфікаційної роботи.....	63

## ВСТУП

На сьогоднішній день фрактальні процеси мають місце у багатьох телекомунікаційних, і не лише телекомунікаційних, системах [1]. Зокрема, в деяких телекомунікаційних системах телетрафік вважається фрактальним процесом [2].

У роботі [2] для деяких систем зроблено припущення, що телетрафік добре описується як стаціонарний неперервний фрактальний процес зі степеневою структурною функцією. Для прогнозу такого телетрафіку у [2] запропоновано використовувати фільтр Колмогорова–Вінера. Подальші роботи з цього напрямку присвячені пошуку вагової функції відповідного фільтру на основі рівняння Ріккати [2,3] та на основі більш загального рівняння Фредгольма [4], причому якщо для рівняння Ріккати можна отримати точний аналітичний розв'язок для вагової функції [3], то рівняння Фредгольма може бути розв'язане лише наближено [4,5].

У роботах [4,5] вагова функція шукалась у вигляді розвинення за деякою системою ортогональних поліномів, та спостерігалась досить дивна поведінка розв'язків в залежності від числа поліномів – точність розв'язку не обов'язково зростала зі збільшенням кількості поліномів. Тим не менш, у досить широкому діапазоні параметрів деякі поліномні розв'язки давали добрий результат. Це є досить природнім, бо, як відомо [6], гарантією збіжності є позитивна визначеність ядра інтегрального рівняння Фредгольма першого роду, а в рамках задачі, що розглядається, ядро не є позитивно визначеним. Проте постає цілком природне питання – а чи не покращиться результат, якщо використати якусь іншу систему поліномів? Поведінка поліномних розв'язків, помічена у [4,5], є майже універсальною, чи інша система поліномів суттєво змінить результати? У роботі [4] використана система поліномів, яка є ортогональною без ваги, у роботі [5] – система поліномів Чебишева другого роду. У даній роботі використовується система поліномів Чебишева першого роду. Метою роботи є отримати відповідну вагову функцію фільтра Колмогорова–Вінера у вигляді розвинення за поліномами Чебишева першого роду та порівняти результати з

результатами робіт [4,5], тобто порівняти отримані розв'язки для різних систем ортогональних поліномів.

Метою економічного розділу є розрахувати трудомісткість та витрати на дослідження фільтру Колмогорова-Вінера.



# 1 СТАН ПИТАННЯ І ПОСТАНОВКА ЗАДАЧІ

## 1.1 Стан питання

### 1.1.1 Загальні відомості про фільтр Колмогорова–Вінера

Фільтр Колмогорова–Вінера є лінійним стаціонарним фільтром, що може розв'язувати дві задачі – задачу виділення корисного сигналу з суми корисного сигналу та шуму, та задачу прогнозування. Так як фільтр Колмогорова–Вінера є лінійним та стаціонарним, то вихід фільтру  $y(t)$  та вхід  $x(t)$  пов'язані співвідношенням

$$y(t) = \int_{-\infty}^{\infty} h(\tau)x(t-\tau)d\tau, \quad (1.1)$$

де  $h(\tau)$  є ваговою функцією фільтра, що через принцип причинності має властивість [7]

$$h(\tau < 0) = 0, \quad (1.2)$$

що дає змогу переписати (1.1) як

$$y(t) = \int_0^{\infty} h(\tau)x(t-\tau)d\tau. \quad (1.3)$$

Вхідний сигнал  $x(t)$  даного фільтру є стаціонарним випадковим процесом, який є сумою корисного сигналу  $s(t)$  та шуму  $n(t)$ :

$$x(t) = n(t) + s(t). \quad (1.4)$$

Задача фільтру є наступною – мінімізувати середньоквадратичне відхилення виходу фільтру  $y(t)$  та корисного сигналу у момент часу  $t + k$ , де  $k$  – часовий інтервал, на який робиться прогноз:

$$\left\langle (y(t) - s(t + k))^2 \right\rangle \rightarrow \min. \quad (1.5)$$

При цьому відомими вважаються автокореляційна функція вхідного сигналу  $R_x(t)$  та взаємна кореляційна функція вхідного сигналу та шуму  $R_{sx}(t)$ . Як відомо [7], для того, щоб виконалось (1.5), вагова функція має задовольняти наступному інтегральному рівнянню:

$$R_{sx}(t + k) = \int_0^{+\infty} d\tau h(\tau) R_x(t - \tau), \quad (1.6)$$

яке має назву інтегрального рівняння Вінера–Хопфа.

Все, що написано до сих пір, стосується неперервних нескінченних у часі процесів. Якщо процеси є дискретними, та вхід фільтра задано лише для моментів часу  $t = 0, 1, 2, \dots, T$ , то рівняння для вагової функції фільтра буде наступним:

$$R_{sx}(t + k) = \sum_{\tau=0}^T h(\tau) R_x(t - \tau), \quad (1.7)$$

а вихід фільтра обчислюється за формулою

$$y(t) = \sum_{\tau=0}^T h(\tau) x(t - \tau). \quad (1.8)$$

У дискретному випадку маємо не неперервну вагову функцію, а набір вагових коефіцієнтів. Фактично, (1.7) є системою лінійних рівнянь на вагові коефіцієнти фільтра.

У випадку неперервних, проте обмежених у часі процесів, маємо наступне. Якщо вхід фільтра – неперервний процес, визначеним при  $t \in [0, T]$ , то маємо наступне рівняння на вагову функцію фільтра:

$$R_{xx}(t+k) = \int_0^t d\tau h(\tau) R_x(t-\tau), \quad (1.9)$$

та вихід фільтра обчислюється за формулою

$$y(t) = \int_0^t h(\tau) x(t-\tau) d\tau. \quad (1.10)$$

Тож для реалізації фільтра Колмогорова–Вінера для неперервних скінчених у часі процесів ми маємо вміти розв’язувати рівняння (1.9).

### 1.1.2 Огляд робіт попередників

Задача прогнозу фрактального трафіку як неперервного стаціонарного процесу на основі фільтра Колмогорова–Вінера була поставлена у [2]. Авторами роботи [2] було помічено, що трафік має фрактальні властивості для мережі IEEE 802.11b та для маршрутизатора між внутрішньою мережею Уфимського державного авіаційного технічного університету та мережею Інтернет. Через велику кількість даних у роботі [2] зазначено, що трафік через велику кількість даних зручно описувати як неперервний у часі процес. Слід зауважити, що фрактальні процеси мають широке використання у телекомунікаційних системах, див. книгу [1] та багаточисельні посилання в ній.

У роботі [2] для опису трафіку запропоновано використовувати модель, в якій його структурна функція  $c(\tau)$  вважається степеневою:

$$c(\tau) \equiv \left\langle (x(t) - x(t - \tau))^2 \right\rangle_t = \alpha \cdot |\tau|^{2H}, \quad (1.11)$$

де  $\alpha$  – деяка стала, та  $H$  – показник Херста, який для процесів, що зберігають свою тенденцію у часі, лежить у діапазоні  $H \in (0,5;1)$ . На основі (1.11) можна отримати наступну кореляційну функцію трафіку:

$$R(t) = \sigma^2 - \frac{\alpha}{2} |t|^{2H}, \quad (1.12)$$

де  $\sigma^2$  – дисперсія відповідного процесу.

У роботі [2] для отримання вагової функції фільтра Колмогорова–Вінера замість рівняння (1.9) було використано більш просте рівняння Ріккати, точний аналітичний розв’язок якого остаточно було отримано у [3]. Проте рівняння Ріккати, яке, можливо, може служити основою для дослідження вагової функції фільтра в деяких частинних випадках, не є застосовним у загальному випадку, тому у роботах [4,5] розглянуто відповідну задачу на основі рівняння (1.9), яке є інтегральним рівнянням Фредгольма першого роду. Таке рівняння навряд чи допускає точний аналітичний розв’язок, тому у роботі [4] воно розв’язується наближено за допомогою розвинення по системі наступних ортогональних поліномів:

$$S_n(\tau) = \frac{S'_n(\tau)}{\sqrt{\int_0^\tau dt (S'_n(t))^2}}, \quad (1.13)$$

де

$$S'_n(\tau) = \begin{vmatrix} \mu_0 & \mu_1 & \mu_2 & \cdots & \mu_n \\ \mu_1 & \mu_2 & \mu_3 & \cdots & \mu_{n+1} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \mu_{n-1} & \mu_n & \mu_{n+1} & \cdots & \mu_{2n-1} \\ 1 & \tau & \tau^2 & \cdots & \tau^n \end{vmatrix}, \quad \mu_n = \int_0^T x^n dx = \frac{T^{n+1}}{n+1}. \quad (1.14)$$

Відповідні поліноми є ортонормованими на проміжку  $t \in [0, T]$  без ваги:

$$\int_0^T dt S_n(t) S_m(t) = \delta_{mn}, \quad (1.15)$$

де  $\delta_{mn}$  – дельта-символ Кронекера:

$$\delta_{mn} = \begin{cases} 1, m = n \\ 0, m \neq n \end{cases}. \quad (1.16)$$

У роботі [5] така ж задача досліджується на основі поліномів Чебишева другого роду. У роботах [4,5] показано, що відповідні обірвані розвинення можуть давати добрі результати у досить широкому діапазоні параметрів, проте наявні і певні недоліки методу. По-перше, таке розвинення не обов'язково дає співпадіння лівої та правої частин інтегрального рівняння – можуть виникати ситуації, коли такі розвинення не працюють. По-друге, точність методу не обов'язково зростає з кількістю поліномів.

Зокрема, у [4] зазначено, що для параметрів

$$T = 100, k = 3, \sigma = 1, 2, H = 0.8, \alpha = \frac{\pi}{2} \quad (1.17)$$

наявна така поведінка розв'язків у наближенні  $n$  поліномів. При  $1 \leq n \leq 8$  точність отриманих розв'язків зростає, та починаючи з наближення трьох поліномів співпадіння лівої та правої частин інтегрального рівняння (1.9) є досить добрим. Проте при  $9 \leq n \leq 15$  метод працює дуже погано – ліва та права частини інтегрального рівняння відрізняються на порядки. Проте при  $16 \leq n \leq 18$  метод знов дуже добре працює – ліва та права частини рівняння (1.9) фактично ідеально співпадають. Дослідження кількості поліномів, більшої ніж 18, стикається з серйозними труднощами через обмежені можливості математичного пакету Wolfram Mathematica.

Слід зауважити, що обрані параметри є дещо «нефізичними», бо для них не виконується нерівність

$$|R(t)| \leq R(0), \quad (1.18)$$

яка у загальному випадку має виконуватись (див., наприклад, [8]). Проте у [4] зазначено, що і в випадку «фізичних» параметрів, тобто таких, які задовольняють (1.18), метод стикається з аналогічними труднощами.

Така поведінка розв'язків може бути пояснена так. Як відомо, гарантією збіжності поліномних розв'язків є те, що ядро відповідного інтегрального рівняння Фредгольму першого роду є позитивно визначеним (див., наприклад, [6]). Проте в задачі, що досліджується, ядром рівняння (1.9) є кореляційна функція  $R(t - \tau)$ , що не є позитивно визначеною. Тому гарантії збіжності поліномних розв'язків нема, хоча у досить широкому діапазоні параметрів такі розв'язки можуть бути досить точними (наприклад, наближення 3–8, 16–18 поліномів для параметрів (1.17)).

У роботі [5] фактично такі ж результати було отримано на основі поліномів Чебишева другого роду – поведінка розв'язків є однаковою як для поліномів Чебишева другого роду, так і для поліномів (1.13).

Можливість розв'язання рівняння (1.9) іншими методами неочевидна. Тож постає наступне питання – а чи не покращить ситуацію використання іншої системи поліномів, ортогональної на відрізку? У даній роботі ставиться мета дослідити цю задачу на основі поліномів Чебишева першого роду.

## **1.2 Постановка задачі**

Постановка задачі до даної кваліфікаційної роботи є наступною: знайти вагову функцію фільтра Колмогорова–Вінера на основі інтегрального рівняння (1.9) з кореляційною функцією (1.12) на основі розвинення по поліномах Чебишева першого роду, ортогональних на відрізку  $t \in [0, T]$ . Дослідити поведінку розв'язків в залежності від кількості поліномів. Порівняти результати з результатами робіт [4,5].

## **1.3 Висновки**

Висновком до першого розділу є те, що задача, поставлена перед даною кваліфікаційною роботою, є актуальною.

## 2 СПЕЦІАЛЬНА ЧАСТИНА

### 2.1 Система поліномів Чебишева, ортогональних на відрізку $t \in [0, T]$

Як відомо [9], поліномами Чебишева першого роду  $T_n(x)$  є наступні поліноми:

$$T_n(x) = \cos(n \arccos x) = \sum_{k=0}^{\left[\frac{n}{2}\right]} C_n^{2k} (x^2 - 1)^k x^{n-2k}, \quad (2.1)$$

де  $\left[\frac{n}{2}\right]$  – ціла частина числа  $\frac{n}{2}$  та

$$C_n^{2k} = \frac{n!}{(2k)!(n-2k)!}. \quad (2.2)$$

Ці поліноми на проміжку  $x \in [-1, 1]$  є ортогональними з вагою  $\frac{1}{\sqrt{1-x^2}}$ :

$$\int_{-1}^1 \frac{T_n(x)T_m(x)}{\sqrt{1-x^2}} dx = A_n \delta_{nm}, \quad (2.3)$$

де

$$A_n = \begin{cases} \pi, & n = 0 \\ \frac{\pi}{2}, & n \neq 0 \end{cases}. \quad (2.4)$$

В рамках нашої задачі необхідно отримати систему поліномів, ортогональну на проміжку  $x \in [0, T]$ , а не на проміжку  $x \in [-1, 1]$ . Тому у формулі (2.2) робимо заміну змінних:



$$A_n \delta_{nm} = \int_{-1}^1 \frac{T_n(x) T_m(x)}{\sqrt{1-x^2}} dx = \left. \begin{array}{l} z = x + 1, dz = dx \\ x = -1 \rightarrow z = 0 \\ x = 1 \rightarrow z = 2 \end{array} \right\} = \int_0^2 \frac{T_n(z-1) T_m(z-1)}{\sqrt{1-(z-1)^2}} dz, \quad (2.5)$$

що на основі ще одної заміни змінних дає

$$A_n \delta_{nm} = \int_0^2 \frac{T_n(z-1) T_m(z-1)}{\sqrt{1-(z-1)^2}} dz = \left. \begin{array}{l} y = \frac{zT}{2}, dy = \frac{T}{2} dz, \\ z = 0 \rightarrow y = 0 \\ z = 2 \rightarrow y = T \end{array} \right\} =$$

$$= \frac{2}{T} \int_0^T \frac{T_n\left(\frac{2y}{T} - 1\right) T_m\left(\frac{2y}{T} - 1\right)}{\sqrt{1 - \left(\frac{2y}{T} - 1\right)^2}} dy. \quad (2.6)$$

Остаточно перепишемо (2.5) у вигляді

$$\int_0^T T_n\left(\frac{2y}{T} - 1\right) T_m\left(\frac{2y}{T} - 1\right) w(y) dy = \frac{T}{2} A_n \delta_{mn} = A'_n \delta_{mn}, \quad (2.7)$$

де

$$A'_n = \begin{cases} \frac{T\pi}{2}, & n = 0 \\ \frac{T\pi}{4}, & n \neq 0 \end{cases}, \quad (2.8)$$

та вагова функція

$$w(y) = \frac{1}{\sqrt{1 - \left(\frac{2y}{T} - 1\right)^2}}. \quad (2.9)$$

Тож, на проміжку система  $t \in [0, T]$  функцій  $T_n\left(\frac{2t}{T} - 1\right)$  є ортогональною з вагою  $w(t)$ , див. формулу (2.9).

Тут і надалі цю систему функцій називатимемо системою поліномів Чебишева, ортогональною на відріжку  $t \in [0, T]$ .

Розглянемо деякі властивості такої системи. З формули (2.1) можна побачити, що поліноми Чебишева з непарними номерами є непарними функціями, а з парними номерами – парними функціями:

$$T_n(x) = \begin{cases} T_n(-x), & n:2 \\ -T_n(-x), & n \neq 2 \end{cases}. \quad (2.10)$$

Як буде показано далі, ця властивість дозволяє значно зменшити кількість обчислень.

## 2.2 Обчислення вагової функції фільтра

У цьому підрозділі на основі інтегрального рівняння (1.9) з кореляційною функцією (1.12) обчислюється вагова функція фільтра Колмогорова–Вінера.

Для розв'язання такої задачі використовується метод, аналогічний тому, який використаний в роботах [4,5]. Вагова функція шукається у вигляді розвинення за поліномами Чебишева, ортогональними на відріжку:

$$h(\tau) = \sum_n g_n T_n\left(\frac{2\tau}{T} - 1\right), \quad (2.11)$$

де  $g_n$  – коефіцієнти при поліномах, які необхідно знайти. На основі рівнянь (1.9) та (2.11) маємо

$$\sum_n g_n \int_0^T d\tau T_n \left( \frac{2\tau}{T} - 1 \right) R(t - \tau) = R(t + k). \quad (2.12)$$

Після множення обох частин рівняння (2.12) на  $T_m \left( \frac{2t}{T} - 1 \right)$  та подальшого інтегрування отримаємо:

$$\sum_n g_n \int_0^T \int_0^T d\tau dt T_n \left( \frac{2\tau}{T} - 1 \right) T_m \left( \frac{2t}{T} - 1 \right) R(t - \tau) = \int_0^T dt T_m \left( \frac{2t}{T} - 1 \right) R(t + k). \quad (2.13)$$

Введемо позначення

$$G_{mn} = \int_0^T \int_0^T d\tau dt T_n \left( \frac{2\tau}{T} - 1 \right) T_m \left( \frac{2t}{T} - 1 \right) R(t - \tau), \quad b_m = \int_0^T dt T_m \left( \frac{2t}{T} - 1 \right) R(t + k), \quad (2.14)$$

де  $G_{mn}$  – так звані інтегральні дужки. Тож рівняння (2.13) переписується як

$$\sum_n g_n G_{mn} = b_m, \quad (2.15)$$

та являє собою нескінченну систему лінійних рівнянь для невідомих коефіцієнтів  $g_n$ , з якою дуже важко працювати. Тому цю систему штучно обривають на скінченній кількості рівнянь:

$$\sum_{n=0}^{l-1} g_n G_{mn} = b_m, \quad m = \overline{0, l-1}. \quad (2.16)$$

З системи рівнянь (2.16) отримуються коефіцієнти  $g_0, g_1, \dots, g_{l-1}$ . Відповідний розв'язок

$$h(\tau) = \sum_{n=0}^{l-1} g_n T_n \left( \frac{2\tau}{T} - 1 \right) \quad (2.17)$$

є розв'язком у наближенні  $l$  поліномів.

Слід зазначити деякі властивості інтегральних дужок (2.14). По-перше, з їх визначення очевидно, що вони є симетричними:

$$G_{mn} = G_{nm}. \quad (2.18)$$

По-друге,

$$\begin{aligned} G_{mn} &= \int_0^T \int_0^T d\tau dt T_n \left( \frac{2\tau}{T} - 1 \right) T_m \left( \frac{2t}{T} - 1 \right) R(t - \tau) = \\ &= \left\{ \begin{array}{l} x = \frac{2\tau}{T} - 1, d\tau = \frac{T}{2} dx, \tau = \frac{T}{2}x + \frac{1}{2}, \tau \in [0, T] \Rightarrow x \in [-1, 1] \\ y = \frac{2t}{T} - 1, dt = \frac{T}{2} dy, t = \frac{T}{2}y + \frac{1}{2}, t \in [0, T] \Rightarrow y \in [-1, 1] \end{array} \right\} = \\ &= \frac{T^2}{4} \int_{-1}^1 \int_{-1}^1 dx dy T_n(x) T_m(y) R \left( \frac{T}{2}y - \frac{T}{2}x \right). \end{aligned} \quad (2.19)$$

Зробивши у (2.19) ще одну заміну змінних, маємо

$$\begin{aligned} G_{mn} &= \frac{T^2}{4} \int_{-1}^1 \int_{-1}^1 dx dy T_n(x) T_m(y) R \left( \frac{T}{2}y - \frac{T}{2}x \right) = \left\{ \begin{array}{l} y = -y' \\ x = -x' \end{array} \right\} = \\ &= \frac{T^2}{4} \int_{-1}^1 \int_{-1}^1 dx' dy' T_n(-x') T_m(-y') R \left( \frac{T}{2}x' - \frac{T}{2}y' \right). \end{aligned} \quad (2.20)$$

На основі того, що кореляційна функція (1.12) парна, звідси маємо

$$G_{mn} = \frac{T^2}{4} \int_{-1}^1 \int_{-1}^1 dx' dy' T_n(-x') T_m(-y') R\left(\frac{T}{2} y' - \frac{T}{2} x'\right). \quad (2.21)$$

Тепер розглянемо випадок, коли  $m$  та  $n$  різної парності. Тоді згідно (2.10) маємо

$$T_n(-x') T_m(-y') = -T_n(x') T_m(y'), \quad (2.22)$$

що з урахуванням (2.21) дає

$$G_{mn} = -\frac{T^2}{4} \int_{-1}^1 \int_{-1}^1 dx' dy' T_n(x') T_m(y') R\left(\frac{T}{2} y' - \frac{T}{2} x'\right). \quad (2.23)$$

Перейменовуючи змінні інтегрування у (2.23), маємо

$$G_{mn} = -\frac{T^2}{4} \int_{-1}^1 \int_{-1}^1 dx dy T_n(x) T_m(y) R\left(\frac{T}{2} y - \frac{T}{2} x\right). \quad (2.24)$$

Порівнюючи (2.19) та (2.24), приходимо до висновку, що

$$G_{mn} = 0 \text{ при } m \text{ та } n \text{ різної парності.} \quad (2.25)$$

На основі властивостей (2.25) та (2.18) бачимо, що не обов'язково безпосередньо обчислювати всі інтегральні дужки. Достатньо обчислити лише ті дужки  $G_{mn}$ , для яких  $m$  та  $n$  однакової парності та  $m \geq n$ . Всі інші дужки можна обчислити на основі формул (2.25) та (2.18), а не обчислювати

безпосередньо, що призводить до значно менших втрат часу при комп'ютерному обчисленні дужок.

Обчислення вагової функції фільтра проводилось у математичному пакеті Wolfram Mathematica 11.0. У роботі обчислення проводилось для наступних параметрів:

$$T = 100; k = 3; \sigma = 1,2; H = 0,8; \alpha = 3 \cdot 10^{-3}. \quad (2.26)$$

По-перше, для них виконується нерівність

$$|R(t)| \leq R(0) \text{ при } t \in [0, T], \quad (2.27)$$

тож вони є «фізичними» у тому сенсі, що теоретично можуть реалізуватись. По-друге, ці параметри використані у роботах [4,5], що дає змогу порівняти результати даної роботи з результатами робіт [4,5]. Дослідження поведінки розв'язків робилось до наближення 18 поліномів включно.

У наближенні  $l$  поліномів вагова функція фільтра Колмогорова–Вінера має вигляд (2.17), де коефіцієнти при поліномах наведено у таблиці 2.1:

Таблиця 2.1 – Чисельні значення коефіцієнтів при поліномах у наближеннях різної кількості поліномів

Кількість поліномів у наближенні	Чисельні значення коефіцієнтів при поліномах, округлені до трьох значущих цифр
1	$g_0 = 4,86 \cdot 10^{-3}$
2	$g_0 = 4,86 \cdot 10^{-3}, g_1 = -2,91 \cdot 10^{-2}$
3	$g_0 = -8,8 \cdot 10^{-2}, g_1 = -2,91 \cdot 10^{-2}, g_2 = -2,06 \cdot 10^{-1}$
4	$g_0 = -8,8 \cdot 10^{-2}, g_1 = -5,01 \cdot 10^{-2}, g_2 = -2,06 \cdot 10^{-1}, g_3 = -3,24 \cdot 10^{-2}$

## Продовження таблиці 2.1

5	$g_0 = 6,61 \cdot 10^{-2}$ , $g_1 = -5,01 \cdot 10^{-2}$ , $g_2 = 1,36 \cdot 10^{-1}$ , $g_3 = -3,24 \cdot 10^{-2}$ , $g_4 = 8,56 \cdot 10^{-2}$
6	$g_0 = 6,61 \cdot 10^{-2}$ , $g_1 = -7,19 \cdot 10^{-2}$ , $g_2 = 1,36 \cdot 10^{-1}$ , $g_3 = -6,02 \cdot 10^{-2}$ , $g_4 = 8,56 \cdot 10^{-2}$ , $g_5 = -3,28 \cdot 10^{-2}$
7	$g_0 = 6,68 \cdot 10^{-2}$ , $g_1 = -7,19 \cdot 10^{-2}$ , $g_2 = 1,39 \cdot 10^{-1}$ , $g_3 = -6,02 \cdot 10^{-2}$ , $g_4 = 1,15 \cdot 10^{-1}$ , $g_5 = -3,28 \cdot 10^{-2}$ , $g_6 = 6,11 \cdot 10^{-2}$
8	$g_0 = 6,68 \cdot 10^{-2}$ , $g_1 = -9,51 \cdot 10^{-2}$ , $g_2 = 1,39 \cdot 10^{-1}$ , $g_3 = -8,71 \cdot 10^{-2}$ , $g_4 = 1,15 \cdot 10^{-1}$ , $g_5 = -6,59 \cdot 10^{-2}$ , $g_6 = 6,11 \cdot 10^{-2}$ , $g_7 = -3,27 \cdot 10^{-2}$
9	$g_0 = 8,02 \cdot 10^{-2}$ , $g_1 = -9,51 \cdot 10^{-2}$ , $g_2 = 1,64 \cdot 10^{-1}$ , $g_3 = -8,71 \cdot 10^{-2}$ , $g_4 = 1,49 \cdot 10^{-1}$ , $g_5 = -6,59 \cdot 10^{-2}$ , $g_6 = 1,11 \cdot 10^{-1}$ , $g_7 = -3,27 \cdot 10^{-2}$ , $g_8 = 5,37 \cdot 10^{-2}$
10	$g_0 = 8,02 \cdot 10^{-2}$ , $g_1 = -1,2 \cdot 10^{-1}$ , $g_2 = 1,64 \cdot 10^{-1}$ , $g_3 = -1,15 \cdot 10^{-1}$ , $g_4 = 1,49 \cdot 10^{-1}$ , $g_5 = -9,86 \cdot 10^{-2}$ , $g_6 = 1,11 \cdot 10^{-1}$ , $g_7 = -6,97 \cdot 10^{-2}$ , $g_8 = 5,37 \cdot 10^{-2}$ , $g_9 = -3,26 \cdot 10^{-2}$
11	$g_0 = 9,53 \cdot 10^{-2}$ , $g_1 = -1,2 \cdot 10^{-1}$ , $g_2 = 1,94 \cdot 10^{-1}$ , $g_3 = -1,15 \cdot 10^{-1}$ , $g_4 = 1,86 \cdot 10^{-1}$ , $g_5 = -9,86 \cdot 10^{-2}$ , $g_6 = 1,57 \cdot 10^{-1}$ , $g_7 = -6,97 \cdot 10^{-2}$ , $g_8 = 1,09 \cdot 10^{-1}$ , $g_9 = -3,26 \cdot 10^{-2}$ , $g_{10} = 5,03 \cdot 10^{-2}$
12	$g_0 = 9,53 \cdot 10^{-2}$ , $g_1 = -1,47 \cdot 10^{-1}$ , $g_2 = 1,94 \cdot 10^{-1}$ , $g_3 = -1,44 \cdot 10^{-1}$ , $g_4 = 1,86 \cdot 10^{-1}$ , $g_5 = -1,32 \cdot 10^{-1}$ , $g_6 = 1,57 \cdot 10^{-1}$ , $g_7 = -1,07 \cdot 10^{-1}$ , $g_8 = 1,09 \cdot 10^{-1}$ , $g_9 = -7,24 \cdot 10^{-2}$ , $g_{10} = 5,03 \cdot 10^{-2}$ , $g_{11} = -3,25 \cdot 10^{-2}$
13	$g_0 = 1,13 \cdot 10^{-1}$ , $g_1 = -1,47 \cdot 10^{-1}$ , $g_2 = 2,29 \cdot 10^{-1}$ , $g_3 = -1,44 \cdot 10^{-1}$ , $g_4 = 2,25 \cdot 10^{-1}$ , $g_5 = -1,32 \cdot 10^{-1}$ , $g_6 = 2,03 \cdot 10^{-1}$ , $g_7 = -1,07 \cdot 10^{-1}$ , $g_8 = 1,64 \cdot 10^{-1}$ , $g_9 = -7,24 \cdot 10^{-2}$ , $g_{10} = 1,09 \cdot 10^{-1}$ , $g_{11} = -3,25 \cdot 10^{-2}$ , $g_{12} = 4,82 \cdot 10^{-2}$

## Продовження таблиці 2.1

14	$g_0 = 1,13 \cdot 10^{-1}$ , $g_1 = -1,76 \cdot 10^{-1}$ , $g_2 = 2,29 \cdot 10^{-1}$ , $g_3 = -1,74 \cdot 10^{-1}$ , $g_4 = 2,25 \cdot 10^{-1}$ , $g_5 = -1,65 \cdot 10^{-1}$ , $g_6 = 2,03 \cdot 10^{-1}$ , $g_7 = -1,45 \cdot 10^{-1}$ , $g_8 = 1,64 \cdot 10^{-1}$ , $g_9 = -1,14 \cdot 10^{-1}$ , $g_{10} = 1,09 \cdot 10^{-1}$ , $g_{11} = -7,44 \cdot 10^{-2}$ , $g_{12} = 4,82 \cdot 10^{-2}$ , $g_{13} = -3,24 \cdot 10^{-2}$
15	$g_0 = 1,31 \cdot 10^{-1}$ , $g_1 = -1,76 \cdot 10^{-1}$ , $g_2 = 2,68 \cdot 10^{-1}$ , $g_3 = -1,74 \cdot 10^{-1}$ , $g_4 = 2,66 \cdot 10^{-1}$ , $g_5 = -1,65 \cdot 10^{-1}$ , $g_6 = 2,50 \cdot 10^{-1}$ , $g_7 = -1,45 \cdot 10^{-1}$ , $g_8 = 2,18 \cdot 10^{-1}$ , $g_9 = -1,14 \cdot 10^{-1}$ , $g_{10} = 1,69 \cdot 10^{-1}$ , $g_{11} = -7,44 \cdot 10^{-2}$ , $g_{12} = 1,09 \cdot 10^{-1}$ , $g_{13} = -3,24 \cdot 10^{-2}$ , $g_{14} = 4,69 \cdot 10^{-2}$ .
16	$g_0 = 1,31 \cdot 10^{-1}$ , $g_1 = -2,07 \cdot 10^{-1}$ , $g_2 = 2,68 \cdot 10^{-1}$ , $g_3 = -2,07 \cdot 10^{-1}$ , $g_4 = 2,66 \cdot 10^{-1}$ , $g_5 = -2,00 \cdot 10^{-1}$ , $g_6 = 2,50 \cdot 10^{-1}$ , $g_7 = -1,84 \cdot 10^{-1}$ , $g_8 = 2,18 \cdot 10^{-1}$ , $g_9 = -1,57 \cdot 10^{-1}$ , $g_{10} = 1,69 \cdot 10^{-1}$ , $g_{11} = -1,20 \cdot 10^{-1}$ , $g_{12} = 1,09 \cdot 10^{-1}$ , $g_{13} = -7,56 \cdot 10^{-2}$ , $g_{14} = 4,69 \cdot 10^{-2}$ , $g_{15} = -3,22 \cdot 10^{-2}$
17	$g_0 = 1,52 \cdot 10^{-1}$ , $g_1 = -2,07 \cdot 10^{-1}$ , $g_2 = 3,09 \cdot 10^{-1}$ , $g_3 = -2,07 \cdot 10^{-1}$ , $g_4 = 3,10 \cdot 10^{-1}$ , $g_5 = -2,00 \cdot 10^{-1}$ , $g_6 = 2,99 \cdot 10^{-1}$ , $g_7 = -1,84 \cdot 10^{-1}$ , $g_8 = 2,72 \cdot 10^{-1}$ , $g_9 = -1,57 \cdot 10^{-1}$ , $g_{10} = 2,30 \cdot 10^{-1}$ , $g_{11} = -1,20 \cdot 10^{-1}$ , $g_{12} = 1,74 \cdot 10^{-1}$ , $g_{13} = -7,56 \cdot 10^{-2}$ , $g_{14} = 1,09 \cdot 10^{-1}$ , $g_{15} = -3,22 \cdot 10^{-2}$ , $g_{16} = 4,59 \cdot 10^{-2}$
18	$g_0 = 1,52 \cdot 10^{-1}$ , $g_1 = -2,40 \cdot 10^{-1}$ , $g_2 = 3,09 \cdot 10^{-1}$ , $g_3 = -2,41 \cdot 10^{-1}$ , $g_4 = 3,10 \cdot 10^{-1}$ , $g_5 = -2,36 \cdot 10^{-1}$ , $g_6 = 2,99 \cdot 10^{-1}$ , $g_7 = -2,23 \cdot 10^{-1}$ , $g_8 = 2,72 \cdot 10^{-1}$ , $g_9 = -2,00 \cdot 10^{-1}$ , $g_{10} = 2,30 \cdot 10^{-1}$ , $g_{11} = -1,66 \cdot 10^{-1}$ , $g_{12} = 1,74 \cdot 10^{-1}$ , $g_{13} = -1,24 \cdot 10^{-1}$ , $g_{14} = 1,09 \cdot 10^{-1}$ , $g_{15} = -7,70 \cdot 10^{-2}$ , $g_{16} = 4,59 \cdot 10^{-2}$ , $g_{17} = -3,21 \cdot 10^{-2}$



Як бачимо, коефіцієнти при кожному конкретному поліномі мають деяку тенденцію до зростання з ростом числа поліномів, проте не зростають швидко зі зростом числа поліномів.

Аналітично перевірити якість отриманих розв'язків, безумовно, дуже важко, тому якість результатів для вагової функції була перевірена побудовою відповідних графіків порівняння лівої та правої частин інтегрального рівняння, розв'язком якого має бути вагова функція фільтра.

На всіх наведених графіках пунктиром позначено ліву частину інтегрального рівняння (1.9) для кореляційної функції (1.12) та чисельних даних (2.26) при наближенні заданої кількості поліномів, а суцільною лінією – праву частину відповідного інтегрального рівняння.

Для наближення одного полінома маємо наступне порівняння:

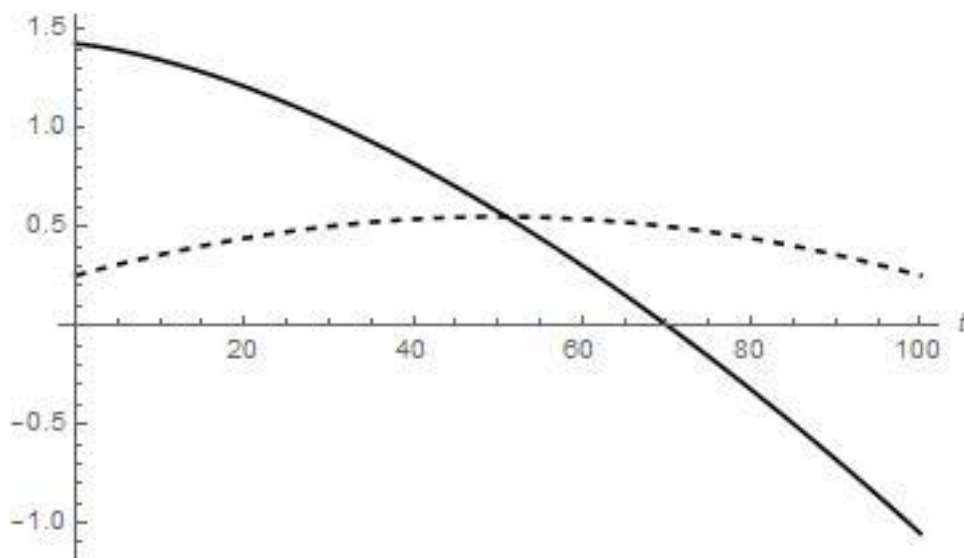


Рисунок 2.1 – чисельне порівняння лівої та правої частини інтегрального рівняння для наближення одного полінома

Як бачимо з рис. 2.1, наближення одного полінома не є точним.

У наближенні двох поліномів маємо наступне:

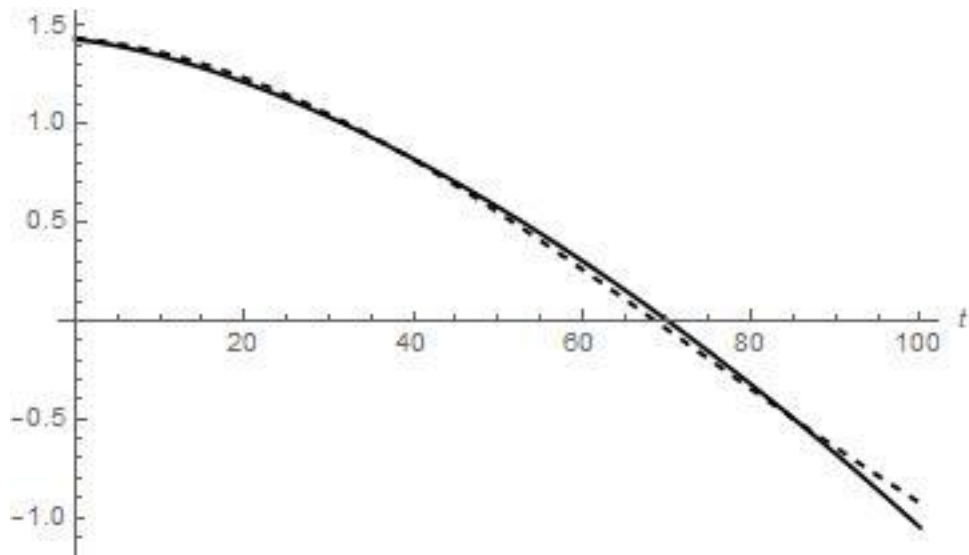


Рисунок 2.2 – чисельне порівняння лівої та правої частини інтегрального рівняння для наближення двох поліномів

Як бачимо з рис. 2.2, наближення двох поліномів є набагато кращим за наближення одного полінома, та досить точним, хоча і має деякі вади при  $t \approx 100$ .

Для наближення трьох поліномів маємо таку ситуацію:

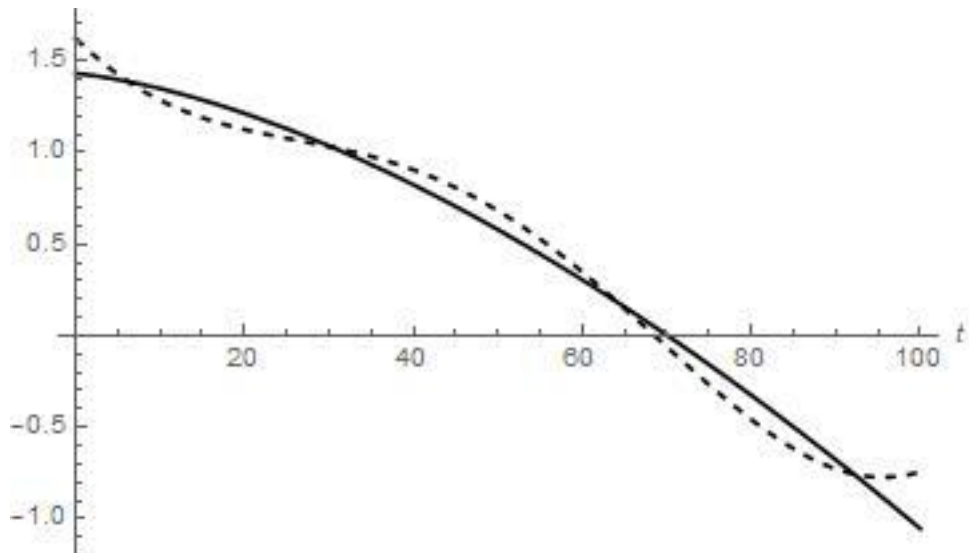


Рисунок 2.3 – порівняння частин лівої та правої частини інтегрального рівняння для наближення трьох поліномів

Як бачимо з рис. 2.3, хоча криві для наближення трьох поліномів є досить близькими, воно є гіршим за наближення двох поліномів. Щодо наближення чотирьох поліномів, маємо

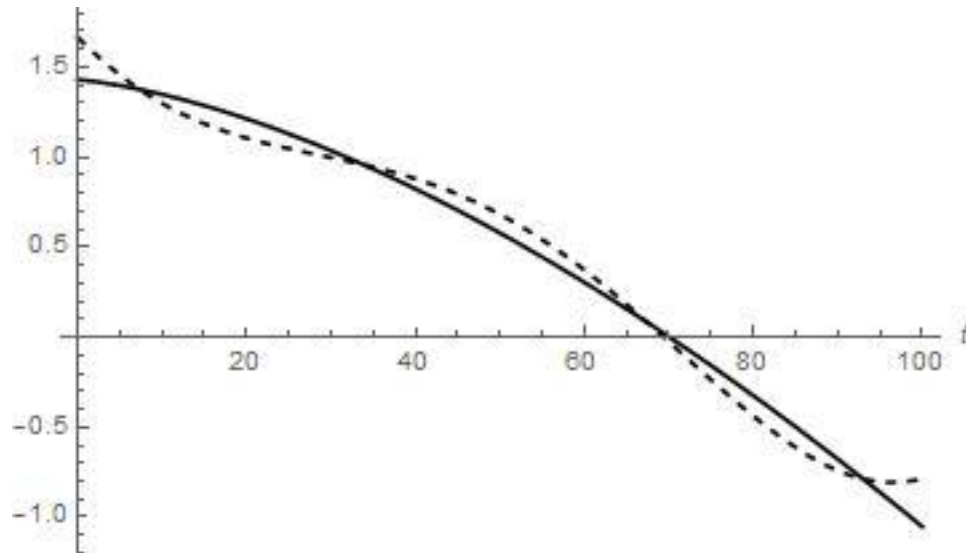


Рисунок 2.4 – чисельне порівняння лівої та правої частини інтегрального рівняння для наближення чотирьох поліномів

Як бачимо з рис. 2.4., наближення чотирьох поліномів є трохи кращим за наближення трьох поліномів, проте гіршим, ніж наближення двох поліномів. У наближенні п'яти поліномів маємо

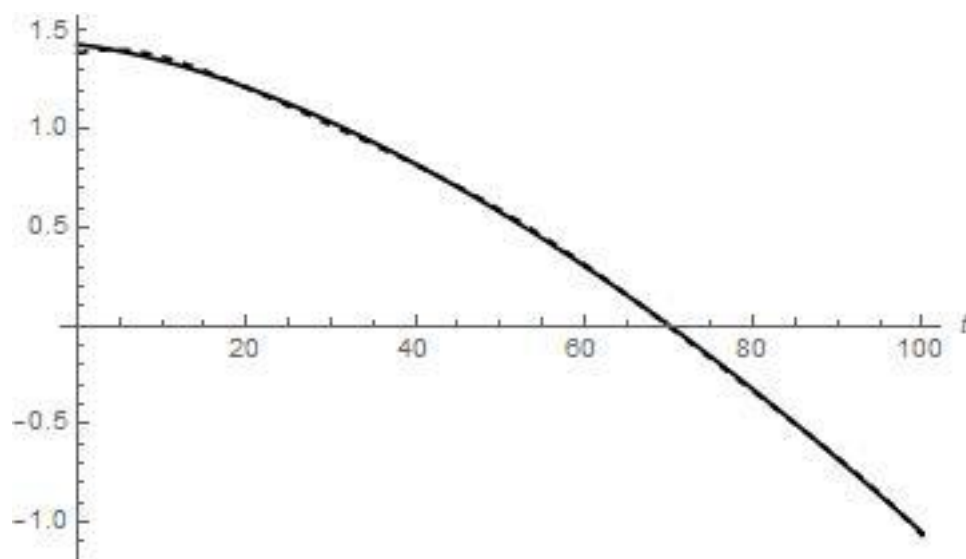


Рисунок 2.5 – чисельне порівняння лівої та правої частини інтегрального рівняння для наближення п'яти поліномів

Як бачимо з рис. 2.5., наближення п'яти поліномів є майже ідеальним, та значно кращим за наближення двох поліномів. Для шести поліномів маємо

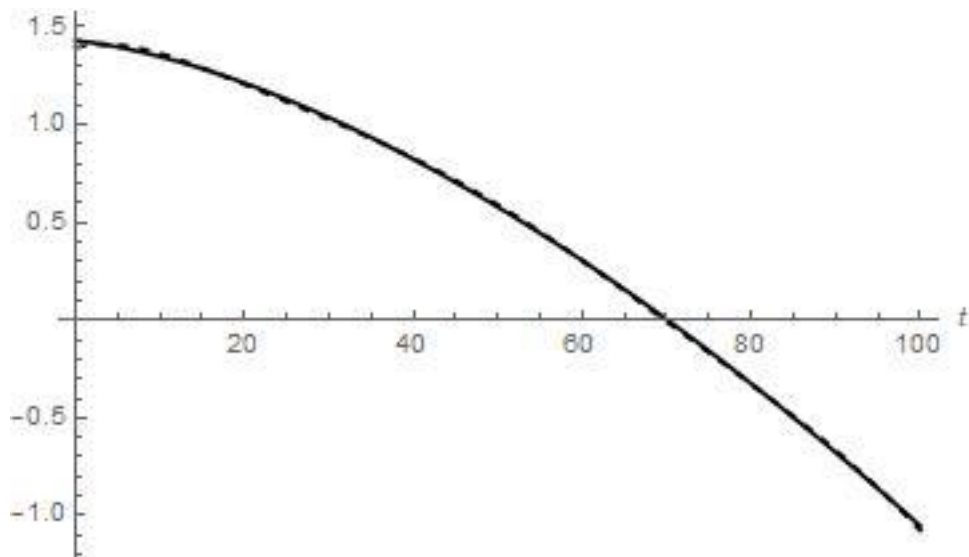


Рисунок 2.6 – чисельне порівняння лівої та правої частини інтегрального рівняння для наближення шести поліномів

Як бачимо з рис. 2.6, наближення шести поліномів є точним та незначно кращим за наближення п'яти поліномів.

Для наближення семи поліномів маємо

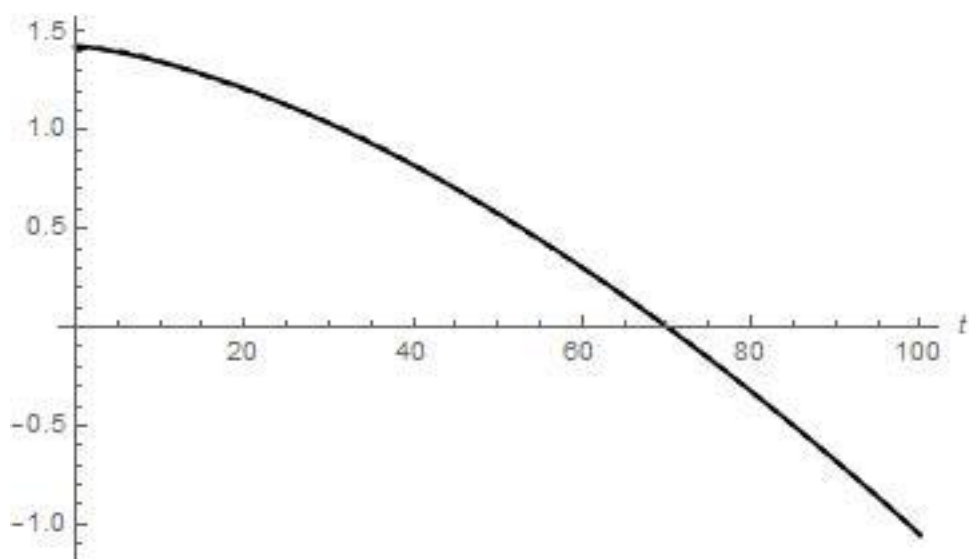


Рисунок 2.7 – чисельне порівняння лівої та правої частини інтегрального рівняння для наближення семи поліномів

Як бачимо з рис. 2.7, наближення семи поліномів є точним та незначно кращим за наближення шести поліномів. Для наближення восьми поліномів маємо

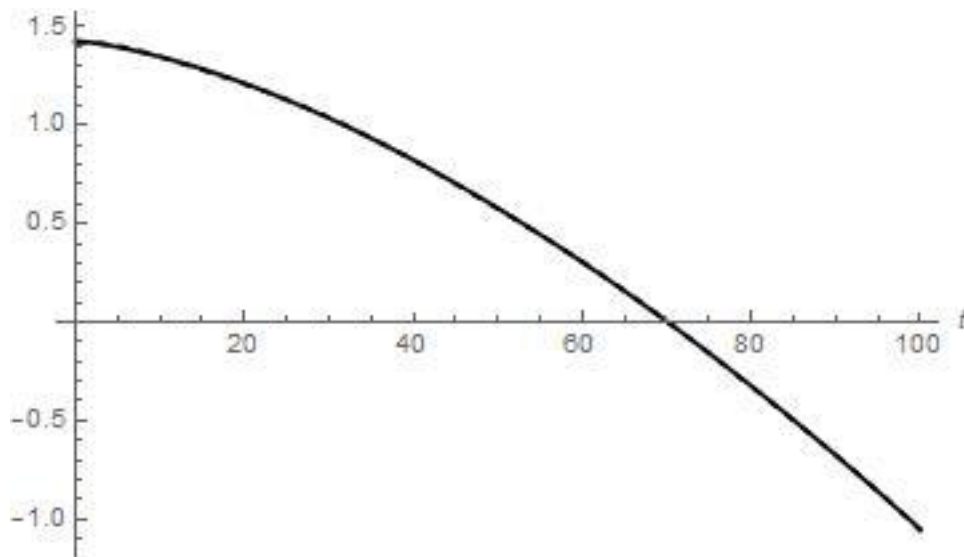


Рисунок 2.8 – чисельне порівняння лівої та правої частини інтегрального рівняння для наближення восьми поліномів

Як бачимо з рис. 2.8, наближення восьми поліномів є точним, та фактично не відчувається різниця між наближеннями семи та восьми поліномів. Для наближення дев'яти поліномів маємо

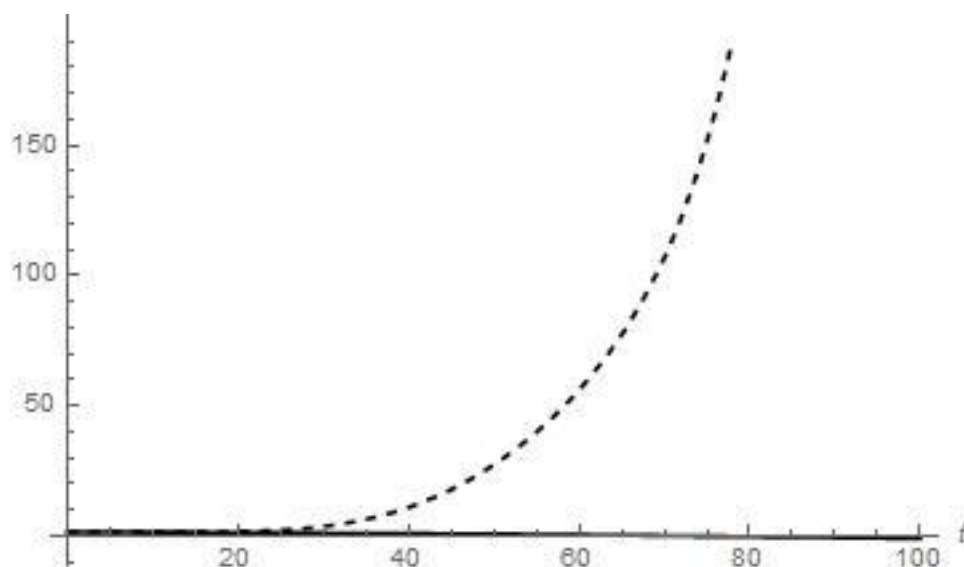


Рисунок 2.9 – чисельне порівняння лівої та правої частини інтегрального рівняння для наближення дев'яти поліномів

Як бачимо з рис. 2.9, для дев'яти поліномів метод працює погано та отримана вагова функція не є розв'язком відповідного інтегрального рівняння. Для десяти поліномів маємо

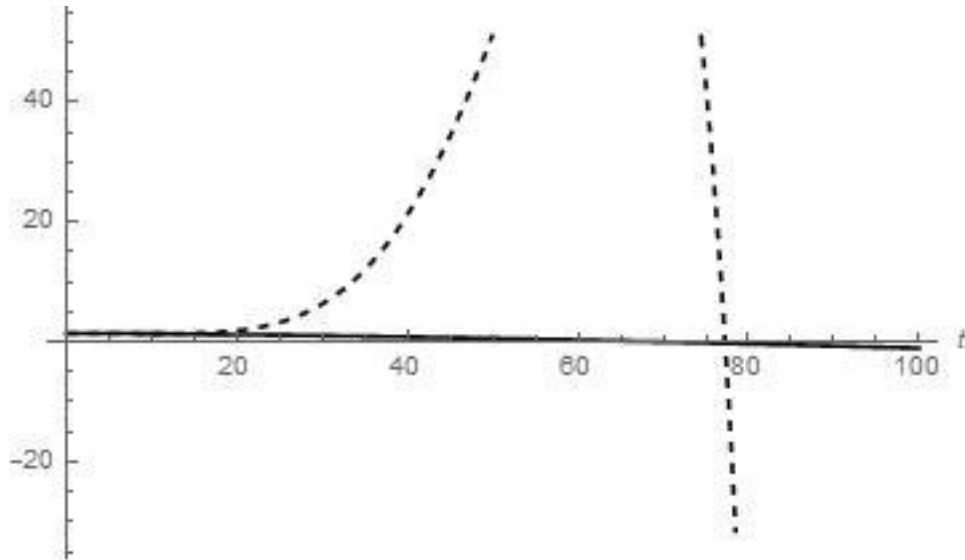


Рисунок 2.10 – чисельне порівняння лівої та правої частини інтегрального рівняння для наближення десяти поліномів

Слід зауважити, що графік лівої частини «вилізає» за область побудови графіку, тому його будуюмо окремо на рис. 2.11:

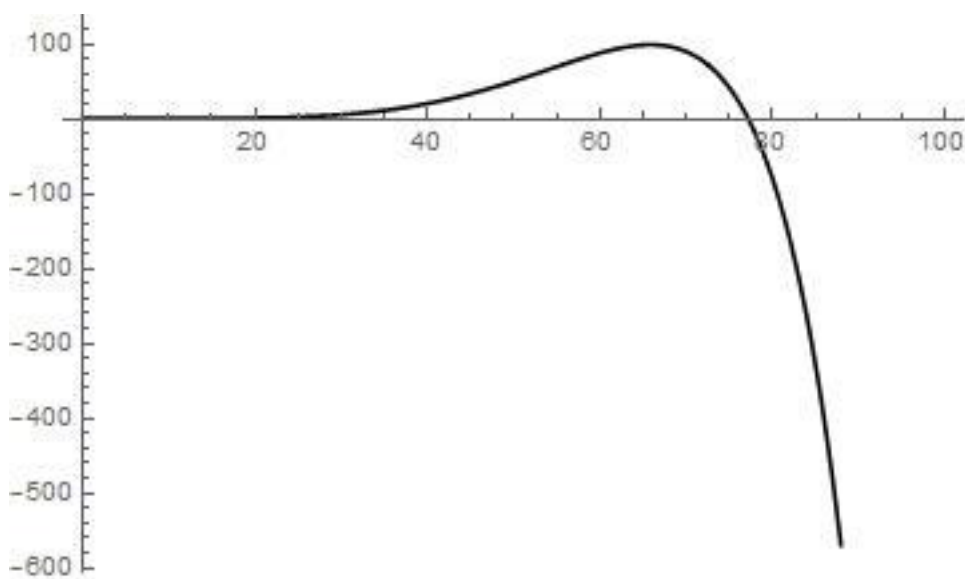


Рисунок 2.11 – ліва частина інтегрального рівняння для наближення десяти поліномів

Як бачимо з рис. 2.10, для десяти поліномів метод теж працює погано. Для одинадцяти поліномів маємо

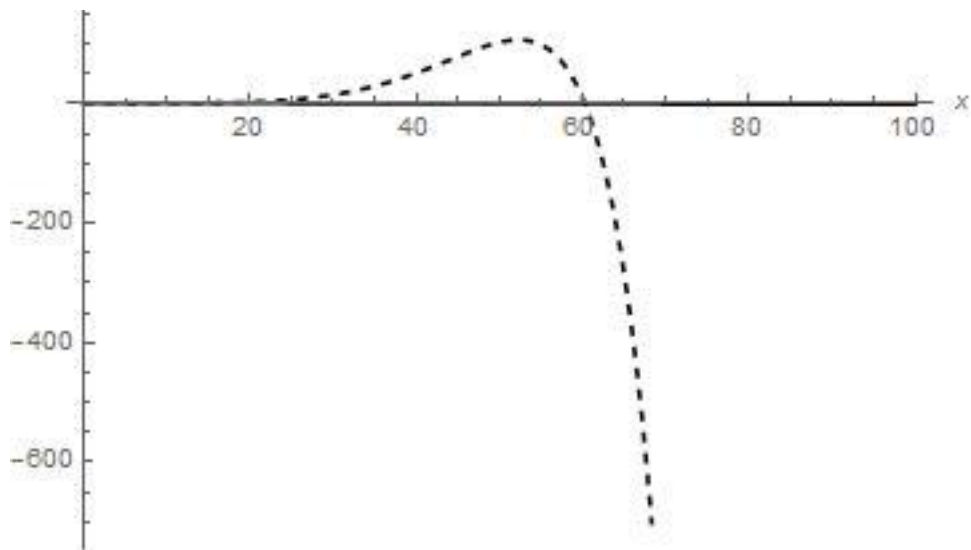


Рисунок 2.12 – чисельне порівняння лівої та правої частини інтегрального рівняння для наближення одинадцяти поліномів

Як бачимо з рис. 2.12, для дванадцяти поліномів метод теж не працює. Для тринадцяти поліномів маємо:

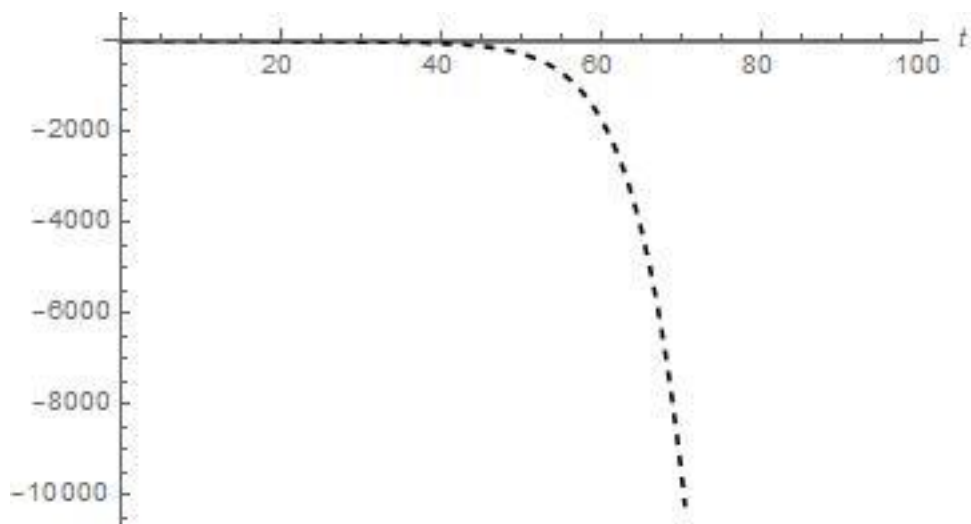


Рисунок 2.13 – чисельне порівняння лівої та правої частини інтегрального рівняння для наближення дванадцяти поліномів

Як бачимо з рис. 2.13, для дванадцяти поліномів метод теж не працює.

Для тринадцяти поліномів маємо:

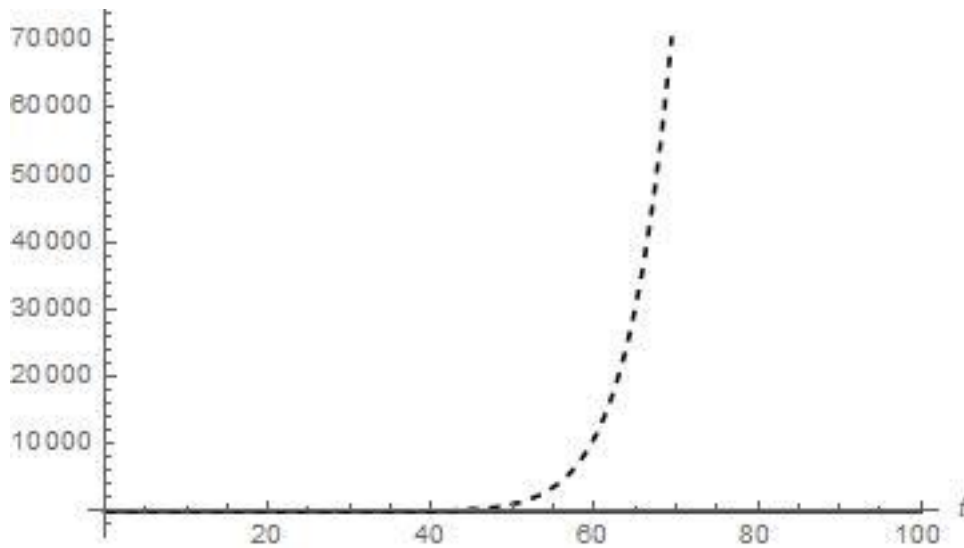


Рисунок 2.14 – чисельне порівняння лівої та правої частини інтегрального рівняння для наближення тринадцяти поліномів

Як бачимо з рис. 2.14, для тринадцяти поліномів метод теж не працює. Для чотирнадцяти поліномів маємо

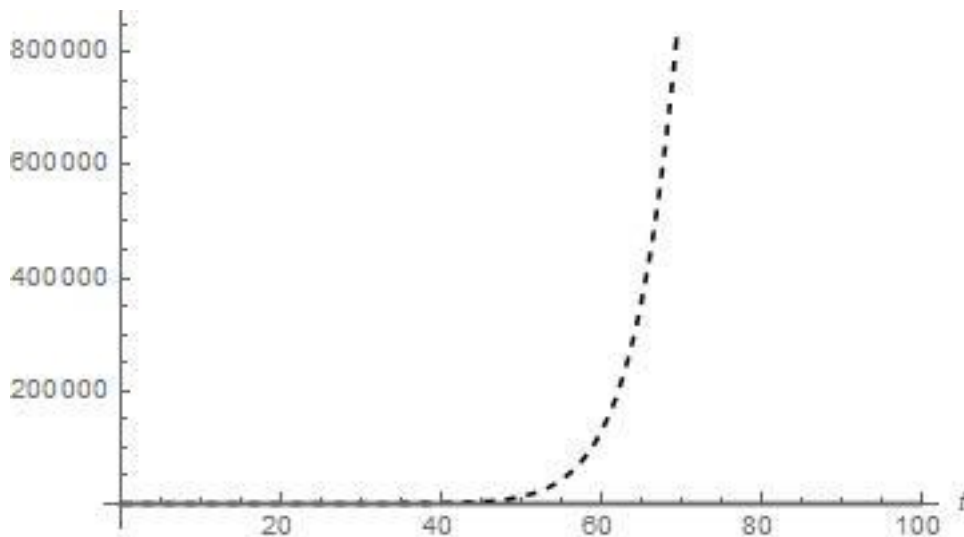


Рисунок 2.15 – чисельне порівняння лівої та правої частини інтегрального рівняння для наближення чотирнадцяти поліномів

Як бачимо з рис. 2.15, для чотирнадцяти поліномів метод теж не працює.



Для п'ятнадцяти поліномів маємо

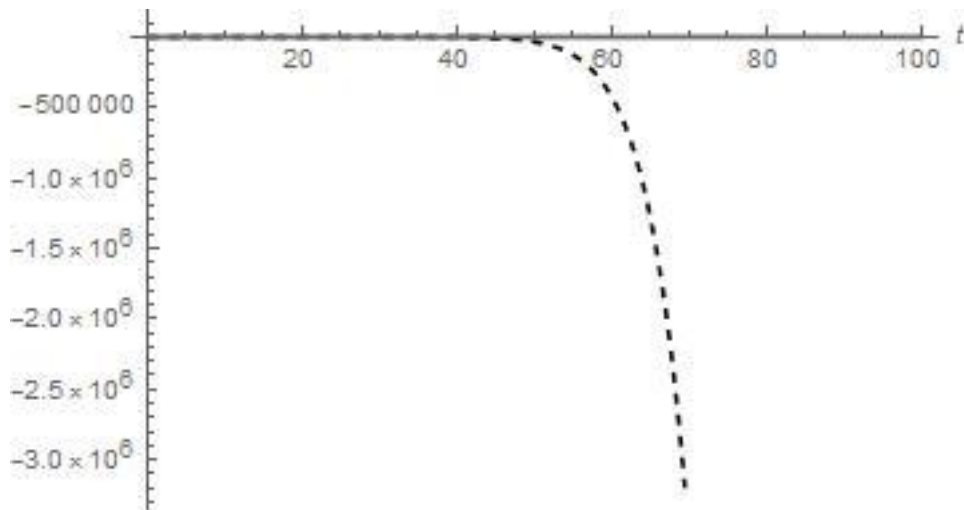


Рисунок 2.16 – чисельне порівняння лівої та правої частини інтегрального рівняння для наближення п'ятнадцяти поліномів

Як бачимо з рис. 2.15, для п'ятнадцяти поліномів метод теж не працює.

Для шістнадцяти поліномів маємо

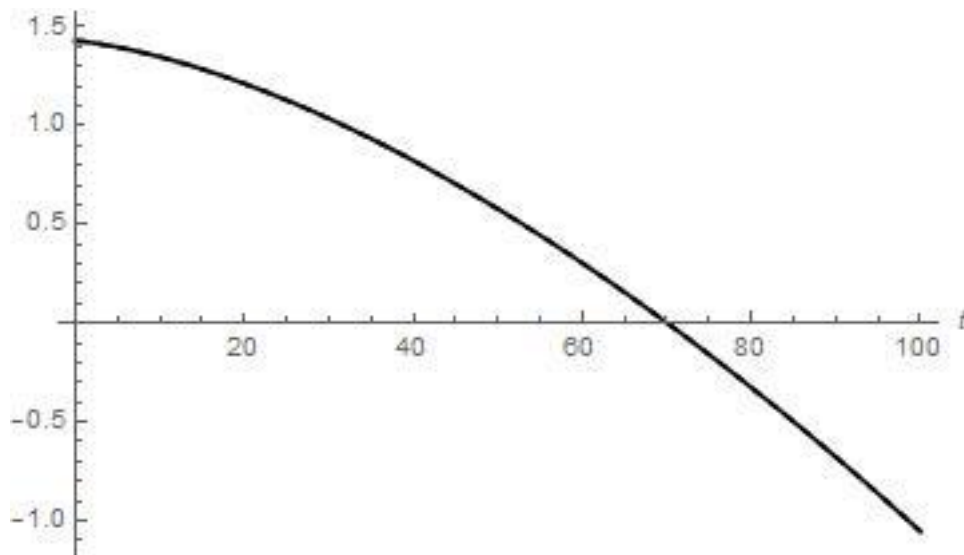


Рисунок 2.17 – чисельне порівняння лівої та правої частини інтегрального рівняння для наближення шістнадцяти поліномів

Як бачимо з рис. 2.17, для шістнадцяти поліномів метод працює фактично ідеально, результати є кращими за наближення восьми поліномів.

Для сімнадцяти поліномів маємо

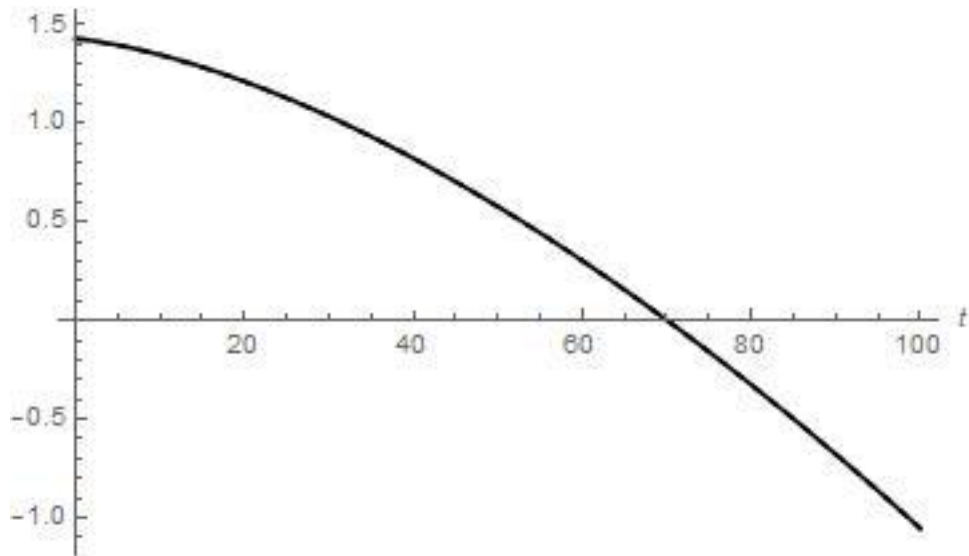


Рисунок 2.18 – чисельне порівняння лівої та правої частини інтегрального рівняння для наближення сімнадцяти поліномів

Як бачимо з рис. 2.18, для сімнадцяти поліномів метод працює фактично ідеально, не відчувається різниця між наближеннями шістнадцяти та сімнадцяти поліномів. Для вісімнадцяти поліномів маємо

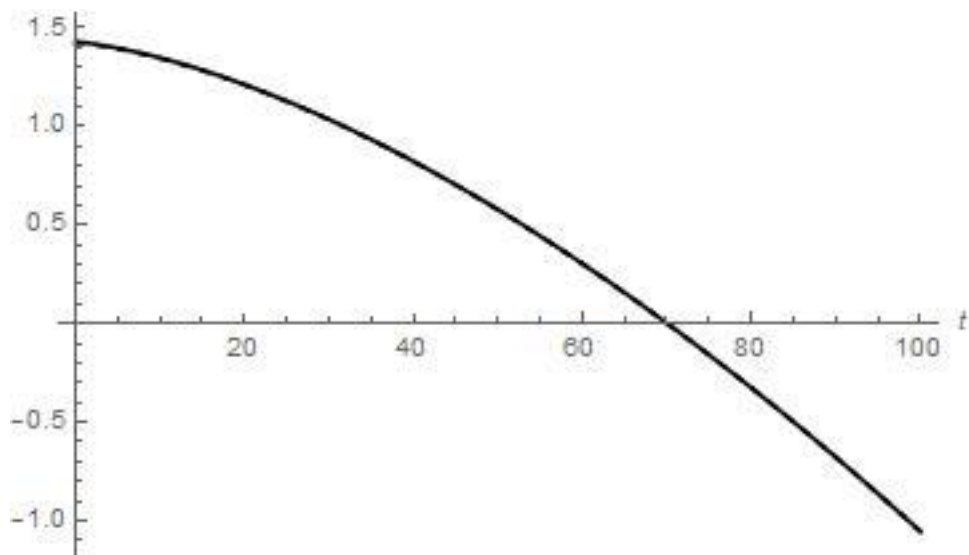


Рисунок 2.19 – чисельне порівняння лівої та правої частини інтегрального рівняння для наближення вісімнадцяти поліномів

Як бачимо з рис. 2.19, для вісімнадцяти поліномів метод працює фактично ідеально, не відчувається різниця між наближеннями сімнадцяти та вісімнадцяти поліномів.

Дослідити наближення більшої кількості поліномів не вдається через те, що на графіках починає з'являтися так зване «миготіння», що свідчить про те, що Wolfram Mathematica 11.0 просто не в змозі адекватно обробляти наближення більшої кількості поліномів.

### **2.3 Порівняння результатів з результатами для інших систем поліномів**

У попередньому підрозділі встановлено наступну поведінку поліномних розв'язків для чисельних параметрів (2.26). Наближення одного полінома не є точним. Наближення двох поліномів є досить точним та набагато кращим за наближення одного полінома. Надалі для наближення трьох поліномів точність методу падає, та лише дещо незначно покращується для наближення чотирьох поліномів.

Проте наближення п'яти поліномів є, фактично, точним – криві лівої та правої частин інтегрального рівняння є дуже близькими для наближення п'яти поліномів. Наближення п'яти поліномів є набагато кращим за наближення двох поліномів. Надалі для наближень шести – восьми поліномів точність методу слабко зростає зі збільшенням кількості поліномів.

Однак наближення дев'яти поліномів дає дуже погані результати – не можна взагалі сказати, що це наближення дає правильний розв'язок інтегрального рівняння. Для цього наближення криві лівої та правої частин інтегрального рівняння відрізняються на порядки. Для наближень десяти – п'ятнадцяти поліномів ситуація навіть погіршується зі зростанням кількості поліномів. Тобто можна казати, що метод, досліджений у цій роботі, не працює для наближень 9–15 поліномів.

Проте наближення шістнадцяти поліномів дає чудові результати – криві лівої та правої частин інтегрального рівняння фактично ідеально співпадають.

Таку ж ситуацію маємо для наближень сімнадцяти та вісімнадцяти поліномів. У роботі [4] дослідження аналогічних розв'язків зроблене для системи поліномів (1.13), а у роботі [5] – на основі поліномів Чебишева другого роду. Поведінка розв'язків, отримана у роботах [4,5] є точно такою же.

Тож не дивлячись на те, що у роботах [4,5] застосовано інші системи поліномів, поведінка розв'язків є, фактично, однаковою для всіх трьох систем поліномів – поліномів (1.13), та поліномів Чебишева першого і другого роду. Тож, скоріш за все, даний тип поведінки не залежить від конкретного вибору системи поліномів.

## 2.4 Висновки

В даній роботі досліджувалась поведінка наближених поліномних розв'язків для вагової функції фільтра Колмогорова–Вінера для фрактальних процесів, які описуються кореляційною функцією (1.12). Дана вагова функція шукалась у вигляді розвинень за поліномами Чебишева першого роду, ортогональними на часовому відрізку, на якому задано вхідний сигнал. Відмінністю роботи від робіт попередників є вибір поліномів Чебишева першого роду, у роботах [4,5] аналогічна задача розв'язувалась на основі інших систем поліномів.

Поведінку розв'язків досліджено для чисельних даних (2.26), для яких виконується нерівність (2.27), тож які мають шанс реалізуватися у реальних системах. Отримано результат, що якісна поведінка розв'язків у наближеннях різної кількості поліномів є ідентичною до відповідної поведінки у роботах [4,5]. Так як параметри (2.26) не є якимись особливими виділеними параметрами, то отримані результати, скоріш за все, означають, що поведінка відповідних розв'язків якісно фактично не залежить від конкретного вибору системи поліномів.

Слід зауважити, що незважаючи на те, що досліджений у роботі метод не є збіжним, тобто його точність не обов'язково зростає зі збільшенням кількості

поліномів, для декількох наближень він дійсно дає хороші результати та майже ідеальне співпадіння лівої та правої частин інтегрального рівняння.

Результати роботи можуть бути застосовані для прогнозування трафіку у телекомунікаційних системах, зокрема таких, як мережі стандарту IEEE 802.11b та для маршрутизаторів між внутрішніми мережами підприємств або університетів та мережею Інтернет.

## 3 ЕКОНОМІЧНИЙ РОЗДІЛ

### 3.1 Розрахунок трудомісткості дослідження фільтру Колмогорова-Вінера

В даному розділі проводяться економічні розрахунки витрат на дослідження фільтру Колмогорова-Вінера. Отримані результати застосовують для прогнозування трафіку у мережах IEEE 802.11b та для маршрутизаторів між внутрішніми мережами підприємств або університетів та мережею Інтернет.

Нормування праці в процесі створення програмного забезпечення утруднене через творчий характер праці програмістів. Тому трудомісткість розробки програмного забезпечення може бути розрахована на основі системи моделей з різною точністю оцінки.

Трудомісткість створення програмного продукту можливо розрахувати за формулою (3.1):

$$t = t_o + t_\mu + t_A + t_\Pi + t_{отл} + t_D, \text{ людино-годин,} \quad (3.1)$$

де  $t_o$  - витрати праці на підготовку і опис поставленого завдання;

$t_\mu$  - витрати праці на дослідження алгоритму рішення задачі;

$t_A$  - витрати праці на розробку блок-схеми алгоритму;

$t_\Pi$  - витрати праці на програмування за розробленою блок-схемою;

$t_{отл}$  - витрати праці на налагодження програми на ЕОМ;

$t_D$  - витрати праці на підготовку документації по завданню.

Складові частини витрат праці визначаються на підставі умовної кількості операторів в програмі. Причому до цього числа входять ті оператори, які програмісту необхідно написати в процесі роботи над завданням з урахуванням можливих уточнень в постановці завдання і удосконалення алгоритму.

Умовна кількість операторів розраховується за формулою (3.2):

$$Q = q \cdot c \cdot (1 + p), \quad (3.2)$$

де  $q$  - передбачувана кількість операторів;

$c$  - коефіцієнт складності програми;

$p$  - коефіцієнт корекції програми в ході її розробки.

Коефіцієнт складності програми  $c$  визначає відносну складність програм завдання по відношенню до типового завдання, складність якого прийнята рівній одиниці. Діапазон його зміни 1,25...2,0.

Коефіцієнт корекції програми  $p$  визначає збільшення об'єму робіт за рахунок внесення змін до алгоритму або програми в результаті уточнення постановки завдання. Величина  $p$  знаходиться в межах 0,05...0,1, що відповідає внесенню 3...5 корекцій, що спричиняють за собою переробку 5-10 % готової програми.

Проведемо розрахунок умовної кількості операторів в програмному продукті, що розробляється:

$$Q = q \cdot c \cdot (1 + p) = 250 \cdot 1 \cdot (1 + 0,1) = 275 \text{ (опер.)} \quad (3.3)$$

Оцінку витрат праці на підготовку і опис завдання приймаємо як  $t_o = 110$  *людино-часів*, тому що дослідження пов'язані з обробкою сигналів в телекомунікаційних системах.

Витрати праці на вивчення опису завдання визначаються з урахуванням уточнення опису і кваліфікації програміста за формулою (3.4):

$$t_{\mu} = \frac{Q \cdot B}{(75...85) \cdot k}, \text{ людино-годин,} \quad (3.4)$$

де  $B$  - коефіцієнт збільшення витрат праці унаслідок недостатнього опису завдання,  $B=1,2...1,5$ ;

$k$  - коефіцієнт кваліфікації програміста, визначуваний залежно від стажу роботи по даній спеціальності.

Він складає при стажі роботи, років:

до 2 - 0,8;

від 2 до 3 - 1,0;

від 3 до 5 - 1,1...1,2;

від 5 до 7 - 1,3...1,4;

Проводимо розрахунок:

$$t_{\mu} = \frac{275 \cdot 1,5}{80 \cdot 1,0} = 5,16 \text{ людино-годин}$$

Витрати праці на розробку алгоритму рішення поставленої задачі:

$$t_A = \frac{Q}{(20...25) \cdot k}, \text{ людино-годин}, \quad (3.5)$$

$$t_A = \frac{275}{25 \cdot 1,0} = 11 \text{ людино-годин}$$

Витрати праці на складання програми по готовій блок-схемі:

$$t_{II} = \frac{Q}{(20...25) \cdot k}, \text{ людино-годин}, \quad (3.6)$$

$$t_{II} = \frac{275}{25 \cdot 1,0} = 11 \text{ людино-годин}$$

Витрати на наладку програми на ЕОМ  $t_{отл}$  розраховуються по наступних формулах:

за умови автономної наладку одного завдання:

$$t_{отл} = \frac{Q}{(4...5) \cdot k}, \text{ людино-годин}, \quad (3.7)$$

$$t_{отл} = \frac{Q}{(4...5) \cdot k} = \frac{275}{5 \cdot 1,0} = 55 \text{ людино-годин}$$



за умови комплексному відкладанню завдання:

$$t_{OTL}^K = 1,5 \cdot t_{OTL}, \text{ людино-годин}, \quad (3.8)$$

$$t_{OTL}^K = 1,5 \cdot t_{OTL} = 1,5 \cdot 55 = 82,5 \text{ людино-годин}$$

Витрати праці на підготовку документації по завданню  $t_D$  визначаються за формулою:

$$t_D = t_{DP} + t_{DO}, \text{ людино-годин}, \quad (3.9)$$

де  $t_{DP}$  - трудомісткість підготовки матеріалів до рукопису, ;

$$t_{DP} = \frac{Q}{(15 \dots 20) \cdot k} = \frac{275}{20 \cdot 1,0} = 13,75$$

$t_{DO}$  - трудомісткість редагування, друку і оформлення документації

$$t_{DO} = 0,75 \cdot t_{DP} = 0,75 \cdot 13,75 = 10,31$$

Отже, трудомісткість створення програмного продукту складає:

$$t_D = t_{DP} + t_{DO} = 13,75 + 10,31 = 24,06 \text{ людино-годин}$$

Таким чином, визначивши трудомісткість окремих показників, розрахуємо сумарну трудомісткість розробки програмного продукту за формулою (3.1):

$$t = 275 + 5,16 + 11 + 11 + 55 + 24,06 = 381,22 \text{ людино-годин}$$

### 3.2 Розрахунок витрат на створення програми

Витрати на створення програми включають витрати на заробітну плату виконавців і вартість машинного часу, необхідного для налагодження програми:

$$K_{IЗ} = Z_{ЗП} + Z_{МЧ}, \text{ грн} \quad (3.10)$$

Заробітна плата виконавців визначається за формулою:

$$Z_{зп} = t \cdot C_{пр}, \text{ грн}, \quad (3.11)$$

де  $t$  - загальна трудомісткість розробки програмного продукту, визначувана по формулі (3.1), людино-годин;

$C_{пр}$  - середня годинна заробітна плата програміста (основна і додаткова) з урахуванням відрахувань на соціальні потреби, грн/год.

Спочатку визначаємо мінімальний щомісячний оклад програміста. Він складає 13000,00 грн. Отже, заробітна плата програміста з урахуванням премій (20%) і можливих надбавок (10%) складає 16900 грн.

Таким чином, річний фонд заробітної плати – 202800 грн. Єдиний соціальний внесок складає 37%, тобто 75036 грн.

Разом, річний фонд заробітної плати з урахуванням відрахувань на соціальні потреби склав 277836 грн.

Визначимо номінальний річний фонд робочого часу, при цьому прийнявши середню тривалість робочого дня рівної 8 годинам:

$$F_H = (T_K - T_{пр} - T_{вих} - T_{отп}) \cdot 8, \text{ годин}, \quad (3.12)$$

де  $T_K$  - кількість календарних днів в році,  $T_K = 365 \text{ днів}$ ;

$T_{пр}$  - кількість святкових днів в році,  $T_{пр} = 10 \text{ днів}$ ;

$T_{вих}$  - кількість вихідних днів в році,  $T_{вих} = 104 \text{ дні}$ ;

$T_{отп}$  - календарна тривалість відпустки  $T_{отп} = 24 \text{ дні}$ .

Отже, річний фонд часу дорівнює:

$$F_H = (365 - 10 - 104 - 24) \cdot 8 = 1816, \text{ годин}$$

Середня годинна заробітна плата програміста визначається співвідношенням:

$$C_{\text{пр}} = \frac{\Phi ЗП_{\text{сн}}}{F_{\text{н}}}, \text{ грн / год}, \quad (3.13)$$

де  $\Phi ЗП_{\text{сн}}$  - річний фонд заробітної плати з урахуванням відрахувань на соціальні потреби;

$F_{\text{н}}$  - річний фонд робочого часу.

$$C_{\text{пр}} = \frac{277836}{1816} = 153 \text{ грн / год}$$

Таким чином, витрати на оплату праці розробника складають:

$$З_{\text{зп}} = t \cdot C_{\text{пр}} = 153 \cdot 381,22 = 58326,66 \text{ грн}$$

Розрахунок вартість машинного часу, необхідного для наладки програми на ЕОМ здійснюється за формулою:

$$З_{\text{мч}} = (t_{\text{отл}} + t_{\text{д}}) \cdot C_{\text{мч}}, \text{ грн}, \quad (3.14)$$

де  $t_{\text{отл}}$  - трудомісткість наладки програми на ЕОМ, яка визначається за формулами (3.7) і (3.8), людино-годин;

$t_{\text{д}}$  - витрати праці на підготовку документація відповідно завданню, яка визначається за формулою (3.9);

$C_{\text{мч}}$  - вартість машино-часу ЕОМ, грн/год.

Для розрахунку вартості машино-часу необхідно знати вартість ЕОМ на момент її придбання і введення в експлуатацію, і вартість споживаної електроенергії. Разом, загальна вартість ЕОМ і потрібного ПО на момент придбання складає 13969 грн. Сума річних амортизаційних відрахувань визначається за формулою (3.15):

$$C_{\text{ам}} = C_{\text{пер}} \cdot H_{\text{а}}, \text{ грн}, \quad (3.15)$$

де  $C_{пер}$  - первинна вартість ЕОМ і необхідного програмного забезпечення;  $H_A$  - норма амортизації,

Для розрахунку амортизаційних відрахування доцільно застосувати метод подвійної амортизації ( $H_A = 2/T$ , де  $T$  – строк експлуатації фондів).  $H_A = 2/5 = 0,5$  чи 50 %.

Таким чином, амортизаційні відрахування у 2019р. склали:

$$CAM_{2019} = 13969 \cdot 0,5 = 6984,5 \text{ грн} \quad (3.16)$$

Отже, залишкова вартість ЕОМ і необхідного програмного забезпечення на складі:

$$\Phi_{ост} = 13969 - 6984,5 = 6984,5 \text{ грн}$$

Розрахунок вартості машино-часу ЕОМ проведемо за формулою:

$$C_{м.ч} = \frac{\Phi_{ост}}{\Phi_{год.раб.ч.}} + W \cdot C_э, \text{ грн/год}, \quad (3.17)$$

де  $\Phi_{год.раб.ч.}$  - річний фонд корисного часу роботи ЕОМ;

$W$  - настановна потужність ЕОМ,  $W = 0,4 \text{ кВт}$ ;

$C_э$  - вартість  $1 \text{ кВт} \cdot \text{год}$  електроенергії.

Річний фонд корисного часу роботи ЕОМ дорівнює річному фонду робочого часу програміста і складає 1816 годин.

Таким чином, вартість машино-години ЕОМ складе:

$$C_{м.ч} = \frac{2787,6}{1816} + 0,4 \cdot 1,82 = 2,26 \text{ грн/год}$$

Проведемо розрахунок вартості машинного часу, необхідного для відладки програми на ЕОМ по формулі (3.14):

$$Z_{MI} = (t_{OGL} + t_D) \cdot C_{MCH} = (82,5 + 24,06) \cdot 2,26 = 240,8 \text{ грн},$$

Отже, витрати на створення програмного продукту складають, виходячи з формули (3.10):

$$K_{IЗ} = 58326,66 + 240,8 = 58567,46 \text{ грн}$$

Визначені таким чином витрати на створення програмного забезпечення є одноразовими капітальними витратами.

### **3.3 Висновки**

Аналіз отриманих результатів показує, що витрати на розробку програмного продукту складають 58567,46 грн. Основна їх частина – це витрати на заробітну плату робітника, що досліджує фільтр Колмогорова-Вінера, яка складає 58326,66 грн.

## ВИСНОВКИ

У роботі розглянуто задачу пошуку вагової функції фільтра Колмогорова–Вінера для неперервних фрактальних процесів. Зокрема, як відомо [2], у деяких телекомунікаційних системах трафік зручно описується як неперервний фрактальний процес, тож дана задача є актуальною задачею телекомунікацій. Відповідний фрактальний процес описано як процес зі степеневою структурною функцією.

Слід зауважити, що дана задача розв'язувалась у роботах [4,5] за допомогою обірваних розвинень по деякій ортогональній системі поліномів. У роботах [4,5] зауважено, що такий метод не обов'язково є збіжним, та показано, що деякі поліномні наближення дають чудові результати, проте для деяких наближень метод взагалі не працює. Скоріш за все, цей факт має місце, бо ядро відповідного інтегрального рівняння не є позитивно визначеним, що, як відомо, не гарантує збіжності методу. Проте поставало питання – як зміняться результати, якщо використати іншу систему поліномів.

На відміну від робіт [4,5], у даній роботі використана інша система поліномів. У роботі [4] використана система поліномів (1.13), у роботі [5] – система поліномів Чебишева другого роду, а у даній роботі – система поліномів Чебишева першого роду, яка є ортогональною з вагою на тому часовому відрізку, на якому визначено вхідний сигнал.

Для чисельних параметрів (2.26) детально досліджено поведінку відповідних поліномних розв'язків у наближеннях від одного до вісімнадцяти поліномів. Наведено всі коефіцієнти при поліномах у даних наближеннях, та для кожного з цих наближень наведено графіки порівняння лівої та правої частин відповідного інтегрального рівняння Фредгольма першого роду.

Отримано наступну поведінку розв'язків. Наближення одного полінома не є точним. Наближення двох поліномів є досить точним, та набагато кращим за наближення одного полінома. Наближення трьох та чотирьох поліномів є близькими між собою, гіршими за наближення двох поліномів, проте кращими за наближення одного поліному. Наближення п'яти – восьми поліномів є майже

ідеальними, кращими за наближення двох поліномів, при цьому точність наближень зростає при збільшенні кількості поліномів від п'яти до восьми. Однак наближення дев'яти – п'ятнадцяти поліномів є дуже поганими – для них метод не працює та ліва частина інтегрального рівняння на порядки відрізняється від правої. Проте наближення шістнадцяти – вісімнадцяти поліномів знов є майже ідеальними. Для дослідження більшої кількості поліномів ресурсів пакету Wolfram Mathematica 11.0 на персональному комп'ютері недостатньо.

Така поведінка розв'язків, безумовно, є досить дивною, проте її можна пояснити тим, що через те, що ядро відповідного інтегрального рівняння не є позитивно визначеним, збіжність методу з зростанням кількості поліномів не є гарантованою.

Дана поведінка систем поліномів для параметрів (2.26) якісно співпадає з отриманою у роботах [4,5] для інших систем поліномів. Чисельні параметри (2.26), для яких досліджено поведінку поліномних розв'язків, не є якимись особливими та виділеними. Тож, скоріш за все, результати роботи свідчать про те, що запропонований метод обірваних розвинень для систем ортогональних поліномів дає ідентичні результати для різних систем поліномів. Скоріш за все, переваги та вади цього методу є перевагами та вадами безпосередньо обраного методу, а не обраних для нього поліномних систем.

У роботах [4,5] зазначено, що для системи поліномів (1.13) даний метод поліномних розвинень може бути застосовним до даної задачі у доволі широкому діапазоні параметрів, хоча його результати слід чисельно перевіряти перед використанням. У цій роботі показано, що скоріш за все поведінка розвинень за поліномами Чебишева першого роду є ідентичною, тож запропонований у даній роботі підхід на основі системи поліномів Чебишева першого роду також може бути застосовним у доволі широкому діапазоні параметрів. Обов'язково слід зауважити, що, зокрема для досліджених даних, декілька поліномних наближень дають добрі результати. Зокрема, для

досліджених даних цими наближеннями є наближення двох, п'яти–восьми та шістнадцяти–вісімнадцяти поліномів.

Результати роботи можуть бути застосовані для прогнозування трафіку у телекомунікаційних системах, зокрема таких, як мережі стандарту IEEE 802.11b та для маршрутизаторів між внутрішніми мережами підприємств або університетів та мережею Інтернет.

В економічному розділі розрахована трудомісткість та витрати на дослідження фільтру Колмогорова-Вінера.



## ПЕРЕЛІК ПОСИЛАНЬ

1. V. Pipiras and M. Taqqu, “Long-Range Dependence and Self-Similarity”, Cambridge University Press, 2017, 668p.
2. В. Х. Багманов, А. М. Комиссаров, А. Х. Султанов, «Прогнозирование телетрафика на основе фрактальных фильтров», Вестник Уфимского государственного авиационного технического университета, т. 9, №6 (24), 2007, с. 217–222.
3. V. N. Gorev, A. Yu. Gusev and V. I. Korniienko, “On the analytical solution of a Volterra integral equation for investigation of fractal processes”, Radio Electronics, Computer Science, Control, 2018, No. 4, p 42–50.
4. V. N. Gorev, A. Yu. Gusev and V. I. Korniienko, “Polynomial solutions for the Kolmogorov–Wiener filter weight function for fractal processes”, Radio Electronics, Computer Science, Control, 2019, No. 2, in press.
5. V. N. Gorev, A. Yu. Gusev and V. I. Korniienko, “Investigation of the Kolmogorov–Wiener filter for treatment of fractal processes on the basis of the Chebyshev polynomials of the second kind”, CEUR Workshop Proceedings, vol. 2353, 2019; online: <http://ceur-ws.org/Vol-2353/paper47.pdf>
6. J. M. Ziman, “Electrons and Phonons. The Theory of Transport Phenomena in Solids”, Oxford University Press, 2001, 576p.
7. W. B. Davenport and W. L. Root, “An Introduction to the Theory of Random Signals and Noise”, Wiley-IEEE Press, 1987, 407 p.
8. І. В. Новицький, С. А. Ус, «Випадкові процеси», Дніпропетровськ, НГУ, 2011, 124 с.
9. I. S. Gradshteyn and I. M. Ryzhik, “Table of Integrals, Series, and Products. Seventh edition”, Translated from the Russian, Translation edited and with a preface by A. Jeffrey and D. Zwillinger, Amsterdam, Elsevier/Academic Press, 2007, 1200p.

**ДОДАТОК А. Відомість матеріалів кваліфікаційної роботи**

<b>№</b>	<b>Формат</b>	<b>Найменування</b>	<b>Кількість листків</b>	<b>Примітки</b>
<i>Документація</i>				
1	A4	Реферат	3	
2	A4	Зміст	1	
3	A4	Вступ	2	
4	A4	Стан питання. Постановка задачі.	7	
5	A4	Спеціальна частина	22	
6	A4	Економічний розділ	8	
7	A4	Висновки	3	
8	A4	Перелік посилань	1	
9	A4	Додаток А	1	
10	A4	Додаток Б	1	
11	A4	Додаток В	10	
12	A4	Додаток Г	1	
13	A4	Додаток Д	1	

**ДОДАТОК Б. Перелік документів на оптичному носії**

- 1 Електронна версія пояснювальної записки  
(ПояснювальнаЗапискаЧеркашин.doc та ПояснювальнаЗапискаЧеркашин.pdf)
- 2 Електронна версія демонстраційного матеріалу  
(ПрезентаціяЧеркашин.pdf)

## ДОДАТОК В. Код у Wolfram Mathematica

### В.1 Розрахунок матриці інтегральних дужок та стовпчику з коефіцієнтів $b_m$

T=100

k=3

$\sigma=1.2$

H=0.8

$\alpha=3*10^{-3}$

$R[x_]=\sigma^2-\alpha/2*(Abs[x])^{2*H}$

G=IdentityMatrix[18]

For[i=1,i<=18,i++,

For[j=1,j<=i,j++,

G[[i,j]]=If[OddQ[i+j],0,Integrate[ChebyshevT[i - 1, (2\*x)/T - 1]\*ChebyshevT[j - 1, (2\*y)/T - 1]\*R[x - y], {x, 0, T}, {y, 0, T}]]

]

]

For[i=1,i<=18,i++,

For[j=1,j<i,j++,

G[[j,i]]=G[[i,j]]

]

]

MatrixForm[G]

B=Range[18]

For[i=1,i<=18,i++, B[[i]]=Integrate[ChebyshevT[i - 1, (2\*x)/T - 1]\*R[x + k], {x, 0, T}]]

MatrixForm[B]

### В. 2 Розрахунок коефіцієнтів при поліномах та побудова відповідних графіків для наближень від одного до вісімнадцяти поліномів

"-----"

```

eN = 18
Ge = IdentityMatrix[eN]
For[i = 1, i <= eN, i++, For[j = 1, j <= eN, j++, Ge[[i,j]] = G[[i,j]]]]
Be = Range[eN]
For[i = 1, i <= eN, i++, Be[[i]] = B[[i]]]
ge = Range[eN]
ge = Inverse[Ge] . Be
MatrixForm[ge]
h[x_] = Sum[ge[[i]]*ChebyshevT[i - 1, (2*x)/T - 1], {i, 1, eN}]
TheLeft[x_] = Integrate[h[y]*(σ^2 - (α/2)*(x - y)^(2*H)), {y, 0, x}] +
Integrate[h[y]*(σ^2 - (α/2)*(y - x)^(2*H)), {y, x, T}]
TheRight[x_] = R[x + k]
Plot[{TheLeft[t], TheRight[t]}, {t, 0, T}, PlotStyle -> {Directive[Black, Dashed],
Black}, AxesLabel -> Automatic, PlotLegends -> "Expressions"]
"-----"

eN = 17
Ge = IdentityMatrix[eN]
For[i = 1, i <= eN, i++, For[j = 1, j <= eN, j++, Ge[[i,j]] = G[[i,j]]]]
Be = Range[eN]
For[i = 1, i <= eN, i++, Be[[i]] = B[[i]]]
ge = Range[eN]
ge = Inverse[Ge] . Be
MatrixForm[ge]
h[x_] = Sum[ge[[i]]*ChebyshevT[i - 1, (2*x)/T - 1], {i, 1, eN}]
TheLeft[x_] = Integrate[h[y]*(σ^2 - (α/2)*(x - y)^(2*H)), {y, 0, x}] +
Integrate[h[y]*(σ^2 - (α/2)*(y - x)^(2*H)), {y, x, T}]
TheRight[x_] = R[x + k]
Plot[{TheLeft[t], TheRight[t]}, {t, 0, T}, PlotStyle -> {Directive[Black, Dashed],
Black}, AxesLabel -> Automatic, PlotLegends -> "Expressions"]
"-----"

```

```

eN = 16
Ge = IdentityMatrix[eN]
For[i = 1, i <= eN, i++, For[j = 1, j <= eN, j++, Ge[[i,j]] = G[[i,j]]]]
Be = Range[eN]
For[i = 1, i <= eN, i++, Be[[i]] = B[[i]]]
ge = Range[eN]
ge = Inverse[Ge] . Be
MatrixForm[ge]
h[x_] = Sum[ge[[i]]*ChebyshevT[i - 1, (2*x)/T - 1], {i, 1, eN}]
TheLeft[x_] = Integrate[h[y]*(σ^2 - (α/2)*(x - y)^(2*H)), {y, 0, x}] +
Integrate[h[y]*(σ^2 - (α/2)*(y - x)^(2*H)), {y, x, T}]
TheRight[x_] = R[x + k]
Plot[{TheLeft[t], TheRight[t]}, {t, 0, T}, PlotStyle -> {Directive[Black, Dashed],
Black}, AxesLabel -> Automatic, PlotLegends -> "Expressions"]
"-----"

eN = 15
Ge = IdentityMatrix[eN]
For[i = 1, i <= eN, i++, For[j = 1, j <= eN, j++, Ge[[i,j]] = G[[i,j]]]]
Be = Range[eN]
For[i = 1, i <= eN, i++, Be[[i]] = B[[i]]]
ge = Range[eN]
ge = Inverse[Ge] . Be
MatrixForm[ge]
h[x_] = Sum[ge[[i]]*ChebyshevT[i - 1, (2*x)/T - 1], {i, 1, eN}]
TheLeft[x_] = Integrate[h[y]*(σ^2 - (α/2)*(x - y)^(2*H)), {y, 0, x}] +
Integrate[h[y]*(σ^2 - (α/2)*(y - x)^(2*H)), {y, x, T}]
TheRight[x_] = R[x + k]
Plot[{TheLeft[t], TheRight[t]}, {t, 0, T}, PlotStyle -> {Directive[Black, Dashed],
Black}, AxesLabel -> Automatic, PlotLegends -> "Expressions"]
"-----"

```

```

eN = 14
Ge = IdentityMatrix[eN]
For[i = 1, i <= eN, i++, For[j = 1, j <= eN, j++, Ge[[i,j]] = G[[i,j]]]]
Be = Range[eN]
For[i = 1, i <= eN, i++, Be[[i]] = B[[i]]]
ge = Range[eN]
ge = Inverse[Ge] . Be
MatrixForm[ge]
h[x_] = Sum[ge[[i]]*ChebyshevT[i - 1, (2*x)/T - 1], {i, 1, eN}]
TheLeft[x_] = Integrate[h[y]*(σ^2 - (α/2)*(x - y)^(2*H)), {y, 0, x}] +
Integrate[h[y]*(σ^2 - (α/2)*(y - x)^(2*H)), {y, x, T}]
TheRight[x_] = R[x + k]
Plot[{TheLeft[t], TheRight[t]}, {t, 0, T}, PlotStyle -> {Directive[Black, Dashed],
Black}, AxesLabel -> Automatic, PlotLegends -> "Expressions"]
"-----"

eN = 13
Ge = IdentityMatrix[eN]
For[i = 1, i <= eN, i++, For[j = 1, j <= eN, j++, Ge[[i,j]] = G[[i,j]]]]
Be = Range[eN]
For[i = 1, i <= eN, i++, Be[[i]] = B[[i]]]
ge = Range[eN]
ge = Inverse[Ge] . Be
MatrixForm[ge]
h[x_] = Sum[ge[[i]]*ChebyshevT[i - 1, (2*x)/T - 1], {i, 1, eN}]
TheLeft[x_] = Integrate[h[y]*(σ^2 - (α/2)*(x - y)^(2*H)), {y, 0, x}] +
Integrate[h[y]*(σ^2 - (α/2)*(y - x)^(2*H)), {y, x, T}]
TheRight[x_] = R[x + k]
Plot[{TheLeft[t], TheRight[t]}, {t, 0, T}, PlotStyle -> {Directive[Black, Dashed],
Black}, AxesLabel -> Automatic, PlotLegends -> "Expressions"]
"-----"

```

```

eN = 12
Ge = IdentityMatrix[eN]
For[i = 1, i <= eN, i++, For[j = 1, j <= eN, j++, Ge[[i,j]] = G[[i,j]]]]
Be = Range[eN]
For[i = 1, i <= eN, i++, Be[[i]] = B[[i]]]
ge = Range[eN]
ge = Inverse[Ge] . Be
MatrixForm[ge]
h[x_] = Sum[ge[[i]]*ChebyshevT[i - 1, (2*x)/T - 1], {i, 1, eN}]
TheLeft[x_] = Integrate[h[y]*(σ^2 - (α/2)*(x - y)^(2*H)), {y, 0, x}] +
Integrate[h[y]*(σ^2 - (α/2)*(y - x)^(2*H)), {y, x, T}]
TheRight[x_] = R[x + k]
Plot[{TheLeft[t], TheRight[t]}, {t, 0, T}, PlotStyle -> {Directive[Black, Dashed],
Black}, AxesLabel -> Automatic, PlotLegends -> "Expressions"]
"-----"

eN = 11
Ge = IdentityMatrix[eN]
For[i = 1, i <= eN, i++, For[j = 1, j <= eN, j++, Ge[[i,j]] = G[[i,j]]]]
Be = Range[eN]
For[i = 1, i <= eN, i++, Be[[i]] = B[[i]]]
ge = Range[eN]
ge = Inverse[Ge] . Be
MatrixForm[ge]
h[x_] = Sum[ge[[i]]*ChebyshevT[i - 1, (2*x)/T - 1], {i, 1, eN}]
TheLeft[x_] = Integrate[h[y]*(σ^2 - (α/2)*(x - y)^(2*H)), {y, 0, x}] +
Integrate[h[y]*(σ^2 - (α/2)*(y - x)^(2*H)), {y, x, T}]
TheRight[x_] = R[x + k]
Plot[{TheLeft[t], TheRight[t]}, {t, 0, T}, PlotStyle -> {Directive[Black, Dashed],
Black}, AxesLabel -> Automatic, PlotLegends -> "Expressions"]
"-----"

```



```

eN = 10
Ge = IdentityMatrix[eN]
For[i = 1, i <= eN, i++, For[j = 1, j <= eN, j++, Ge[[i,j]] = G[[i,j]]]]
Be = Range[eN]
For[i = 1, i <= eN, i++, Be[[i]] = B[[i]]]
ge = Range[eN]
ge = Inverse[Ge] . Be
MatrixForm[ge]
h[x_] = Sum[ge[[i]]*ChebyshevT[i - 1, (2*x)/T - 1], {i, 1, eN}]
TheLeft[x_] = Integrate[h[y]*(σ^2 - (α/2)*(x - y)^(2*H)), {y, 0, x}] +
Integrate[h[y]*(σ^2 - (α/2)*(y - x)^(2*H)), {y, x, T}]
TheRight[x_] = R[x + k]
Plot[{TheLeft[t], TheRight[t]}, {t, 0, T}, PlotStyle -> {Directive[Black, Dashed],
Black}, AxesLabel -> Automatic, PlotLegends -> "Expressions"]
Plot[{TheLeft[x]}, {x, 0, T}, PlotStyle -> {Black}, PlotLegends -> "Expressions"]
"-----"
eN = 9
Ge = IdentityMatrix[eN]
For[i = 1, i <= eN, i++, For[j = 1, j <= eN, j++, Ge[[i,j]] = G[[i,j]]]]
Be = Range[eN]
For[i = 1, i <= eN, i++, Be[[i]] = B[[i]]]
ge = Range[eN]
ge = Inverse[Ge] . Be
MatrixForm[ge]
h[x_] = Sum[ge[[i]]*ChebyshevT[i - 1, (2*x)/T - 1], {i, 1, eN}]
TheLeft[x_] = Integrate[h[y]*(σ^2 - (α/2)*(x - y)^(2*H)), {y, 0, x}] +
Integrate[h[y]*(σ^2 - (α/2)*(y - x)^(2*H)), {y, x, T}]
TheRight[x_] = R[x + k]
Plot[{TheLeft[t], TheRight[t]}, {t, 0, T}, PlotStyle -> {Directive[Black, Dashed],
Black}, AxesLabel -> Automatic, PlotLegends -> "Expressions"]

```

```

"-----"
eN = 8
Ge = IdentityMatrix[eN]
For[i = 1, i <= eN, i++, For[j = 1, j <= eN, j++, Ge[[i,j]] = G[[i,j]]]]
Be = Range[eN]
For[i = 1, i <= eN, i++, Be[[i]] = B[[i]]]
ge = Range[eN]
ge = Inverse[Ge] . Be
MatrixForm[ge]
h[x_] = Sum[ge[[i]]*ChebyshevT[i - 1, (2*x)/T - 1], {i, 1, eN}]
TheLeft[x_] = Integrate[h[y]*(σ^2 - (α/2)*(x - y)^(2*H)), {y, 0, x}] +
Integrate[h[y]*(σ^2 - (α/2)*(y - x)^(2*H)), {y, x, T}]
TheRight[x_] = R[x + k]
Plot[{TheLeft[t], TheRight[t]}, {t, 0, T}, PlotStyle -> {Directive[Black, Dashed],
Black}, AxesLabel -> Automatic, PlotLegends -> "Expressions"]
"-----"

eN = 7
Ge = IdentityMatrix[eN]
For[i = 1, i <= eN, i++, For[j = 1, j <= eN, j++, Ge[[i,j]] = G[[i,j]]]]
Be = Range[eN]
For[i = 1, i <= eN, i++, Be[[i]] = B[[i]]]
ge = Range[eN]
ge = Inverse[Ge] . Be
MatrixForm[ge]
h[x_] = Sum[ge[[i]]*ChebyshevT[i - 1, (2*x)/T - 1], {i, 1, eN}]
TheLeft[x_] = Integrate[h[y]*(σ^2 - (α/2)*(x - y)^(2*H)), {y, 0, x}] +
Integrate[h[y]*(σ^2 - (α/2)*(y - x)^(2*H)), {y, x, T}]
TheRight[x_] = R[x + k]
Plot[{TheLeft[t], TheRight[t]}, {t, 0, T}, PlotStyle -> {Directive[Black, Dashed],
Black}, AxesLabel -> Automatic, PlotLegends -> "Expressions"]

```

```

"-----"
eN = 6
Ge = IdentityMatrix[eN]
For[i = 1, i <= eN, i++, For[j = 1, j <= eN, j++, Ge[[i,j]] = G[[i,j]]]]
Be = Range[eN]
For[i = 1, i <= eN, i++, Be[[i]] = B[[i]]]
ge = Range[eN]
ge = Inverse[Ge] . Be
MatrixForm[ge]
h[x_] = Sum[ge[[i]]*ChebyshevT[i - 1, (2*x)/T - 1], {i, 1, eN}]
TheLeft[x_] = Integrate[h[y]*(σ^2 - (α/2)*(x - y)^(2*H)), {y, 0, x}] +
Integrate[h[y]*(σ^2 - (α/2)*(y - x)^(2*H)), {y, x, T}]
TheRight[x_] = R[x + k]
Plot[{TheLeft[t], TheRight[t]}, {t, 0, T}, PlotStyle -> {Directive[Black, Dashed],
Black}, AxesLabel -> Automatic, PlotLegends -> "Expressions"]
"-----"

eN = 5
Ge = IdentityMatrix[eN]
For[i = 1, i <= eN, i++, For[j = 1, j <= eN, j++, Ge[[i,j]] = G[[i,j]]]]
Be = Range[eN]
For[i = 1, i <= eN, i++, Be[[i]] = B[[i]]]
ge = Range[eN]
ge = Inverse[Ge] . Be
MatrixForm[ge]
h[x_] = Sum[ge[[i]]*ChebyshevT[i - 1, (2*x)/T - 1], {i, 1, eN}]
TheLeft[x_] = Integrate[h[y]*(σ^2 - (α/2)*(x - y)^(2*H)), {y, 0, x}] +
Integrate[h[y]*(σ^2 - (α/2)*(y - x)^(2*H)), {y, x, T}]
TheRight[x_] = R[x + k]
Plot[{TheLeft[t], TheRight[t]}, {t, 0, T}, PlotStyle -> {Directive[Black, Dashed],
Black}, AxesLabel -> Automatic, PlotLegends -> "Expressions"]

```

```

"-----"
eN = 4
Ge = IdentityMatrix[eN]
For[i = 1, i <= eN, i++, For[j = 1, j <= eN, j++, Ge[[i,j]] = G[[i,j]]]]
Be = Range[eN]
For[i = 1, i <= eN, i++, Be[[i]] = B[[i]]]
ge = Range[eN]
ge = Inverse[Ge] . Be
MatrixForm[ge]
h[x_] = Sum[ge[[i]]*ChebyshevT[i - 1, (2*x)/T - 1], {i, 1, eN}]
TheLeft[x_] = Integrate[h[y]*(σ^2 - (α/2)*(x - y)^(2*H)), {y, 0, x}] +
Integrate[h[y]*(σ^2 - (α/2)*(y - x)^(2*H)), {y, x, T}]
TheRight[x_] = R[x + k]
Plot[{TheLeft[t], TheRight[t]}, {t, 0, T}, PlotStyle -> {Directive[Black, Dashed],
Black}, AxesLabel -> Automatic, PlotLegends -> "Expressions"]
"-----"

eN = 3
Ge = IdentityMatrix[eN]
For[i = 1, i <= eN, i++, For[j = 1, j <= eN, j++, Ge[[i,j]] = G[[i,j]]]]
Be = Range[eN]
For[i = 1, i <= eN, i++, Be[[i]] = B[[i]]]
ge = Range[eN]
ge = Inverse[Ge] . Be
MatrixForm[ge]
h[x_] = Sum[ge[[i]]*ChebyshevT[i - 1, (2*x)/T - 1], {i, 1, eN}]
TheLeft[x_] = Integrate[h[y]*(σ^2 - (α/2)*(x - y)^(2*H)), {y, 0, x}] +
Integrate[h[y]*(σ^2 - (α/2)*(y - x)^(2*H)), {y, x, T}]
TheRight[x_] = R[x + k]
Plot[{TheLeft[t], TheRight[t]}, {t, 0, T}, PlotStyle -> {Directive[Black, Dashed],
Black}, AxesLabel -> Automatic, PlotLegends -> "Expressions"]

```

```

"-----"
eN = 2
Ge = IdentityMatrix[eN]
For[i = 1, i <= eN, i++, For[j = 1, j <= eN, j++, Ge[[i,j]] = G[[i,j]]]]
Be = Range[eN]
For[i = 1, i <= eN, i++, Be[[i]] = B[[i]]]
ge = Range[eN]
ge = Inverse[Ge] . Be
MatrixForm[ge]
h[x_] = Sum[ge[[i]]*ChebyshevT[i - 1, (2*x)/T - 1], {i, 1, eN}]
TheLeft[x_] = Integrate[h[y]*(σ^2 - (α/2)*(x - y)^(2*H)), {y, 0, x}] +
Integrate[h[y]*(σ^2 - (α/2)*(y - x)^(2*H)), {y, x, T}]
TheRight[x_] = R[x + k]
Plot[{TheLeft[t], TheRight[t]}, {t, 0, T}, PlotStyle -> {Directive[Black, Dashed],
Black}, AxesLabel -> Automatic, PlotLegends -> "Expressions"]
"-----"

eN = 1
Ge = IdentityMatrix[eN]
For[i = 1, i <= eN, i++, For[j = 1, j <= eN, j++, Ge[[i,j]] = G[[i,j]]]]
Be = Range[eN]
For[i = 1, i <= eN, i++, Be[[i]] = B[[i]]]
ge = Range[eN]
ge = Inverse[Ge] . Be
MatrixForm[ge]
h[x_] = Sum[ge[[i]]*ChebyshevT[i - 1, (2*x)/T - 1], {i, 1, eN}]
TheLeft[x_] = Integrate[h[y]*(σ^2 - (α/2)*(x - y)^(2*H)), {y, 0, x}] +
Integrate[h[y]*(σ^2 - (α/2)*(y - x)^(2*H)), {y, x, T}]
TheRight[x_] = R[x + k]
Plot[{TheLeft[t], TheRight[t]}, {t, 0, T}, PlotStyle -> {Directive[Black, Dashed],
Black}, AxesLabel -> Automatic, PlotLegends -> "Expressions"]

```

**ДОДАТОК Г. Відгук керівника економічного розділу  
ВІДГУК**

Керівник розділу

\_\_\_\_\_

(підпис)

Н. М. Романюк

**ДОДАТОК Д. Відгук керівника кваліфікаційної роботи****Відгук на кваліфікаційну роботу****студента групи 172-16зск-1 Черкашина С. Б.****на тему: «Розробка імпульсної характеристики Колмогорова–Вінера для неперервних фрактальних процесів в телекомунікаційних мережах»**

Пояснювальна записка розташована на 63 сторінках і містить, у тому числі, вступ, три розділи та висновки. Метою роботи є дослідження вагової функції фільтра Колмогорова–Вінера для фрактальних процесів на основі поліномів Чебишева першого роду та порівняння результатів для різних систем поліномів.

Рівень запозичень у кваліфікаційній роботі відповідає вимогам «Положення про систему виявлення та запобігання плагіату».

Серед позитивних сторін роботи слід відмітити, зокрема, те, що при виконанні роботи С. Б. Черкашин продемонстрував добрий рівень знань на навичок. У роботі досліджено поведінку відповідних розв'язків та показано, що результати поліномних розвинень є практично ідентичними для різних систем поліномів. Зазначено, що результати можуть бути застосовними, зокрема, для прогнозування трафіку у мережах IEEE 802.11b та для маршрутизаторів між внутрішніми мережами підприємств або університетів та мережею Інтернет.

Серед недоліків роботи слід відмітити, зокрема незначні стилістичні недоліки. Також, можливо, слід було дослідити поведінку відповідних розв'язків для ще декількох наборів параметрів.

Незважаючи на недоліки, в цілому робота задовольняє усім вимогам, що висуваються до кваліфікаційних робіт бакалаврів, заслуговує оцінки «відмінно», а її автор Черкашин С.Б. – присвоєння кваліфікації 3114 «Технічний фахівець в галузі електроніки та телекомунікацій» за спеціальністю 172 «Телекомунікації та радіотехніка».

**Керівник роботи, д.т.н., зав. каф. БІТ****В. І. Корнієнко**