

Міністерство освіти і науки України
Сумський державний університет
Факультет технічних систем та енергоефективних технологій
Кафедра комп'ютерної механіки імені Володимира Марцинковського

КВАЛІФІКАЦІЙНА РОБОТА

бакалавра

на тему

**Використання системи комп'ютерного моделювання MATLAB
для обробки і аналізу діагностичної інформації**

Виконав: студент 4 курсу групи КМ–81/1
Лучка Р.І.

Рівень підготовки: бакалавр

Спеціальність: 131 Прикладна механіка

ОПП: Комп'ютерний інжиніринг в
механіці

Керівник: доцент Савченко Є.М.

Рецензент: зав. кафедри Загорулько А.В.

РЕФЕРАТ

Кваліфікаційна робота бакалавра: 60 с., 19 рис., 15 дж.

Мета роботи: дослідження можливостей системи комп'ютерного моделювання MATLAB для обробки і аналізу діагностичної інформації про технічний стан роторних машин.

Об'єкт дослідження: динамічні процеси у роторних машинах.

Предмет дослідження: можливості системи комп'ютерного моделювання MATLAB для обробки і аналізу діагностичної інформації.

Методи дослідження: комп'ютерне моделювання за допомогою програмного комплексу MATLAB.

Актуальність теми полягає у вдосконаленні існуючих та розробці нових методів та засобів обробки і аналізу діагностичної інформації про технічний стан роторних машин на основі використання можливостей сучасних програмних комплексів, таких як MATLAB.

В роботі показано, що віброакустична діагностика є найбільш прийнятною для оцінки технічного стану роторних машин та механізмів. Вона найбільш повно відображає природу явищ, робочих процесів, що протікають в них і використовує як джерело інформації вібросигнал, який має велику інформаційну ємність, малу інерційність, велику швидкість поширення, широкий частотний і динамічний діапазони. Ці якості вібросигналу дозволяють оперативно, а при використанні ЕОМ і в реальному масштабі часу, оцінювати технічний стан об'єкта діагностування. MATLAB є ідеальним програмним засобом для створення систем вимірювання, а також систем автоматизації управління на основі технології віртуальних приладів. MATLAB-програма в комплексі з такими апаратними засобами, як багатоканальні вимірювальні аналого-цифрові перетворювачі, дозволяє розробляти системи вимірювання, контролю, діагностики та управління практично будь-якої складності.

КОМП'ЮТЕРНЕ МОДЕЛЮВАННЯ, MATLAB, РОТОРНІ МАШИНИ,
ДІАГНОСТИЧНА ІНФОРМАЦІЯ, ОБРОБКА, АНАЛІЗ

ЗМІСТ

ВСТУП	5
1 ВИКОРИСТАННЯ СИСТЕМИ КОМП'ЮТЕРНОГО МОДЕЛЮВАННЯ MATLAB ДЛЯ ОБРОБКИ І АНАЛІЗУ ДІАГНОСТИЧНОЇ ІНФОРМАЦІЇ	7
1.1 Система MATLAB і її можливості по обробці сигналів	7
1.2 Програма Signal Processing Tool (SPTool): головне вікно	9
1.3 Завантаження сигналу	10
1.4 Перегляд графіка сигналу	11
1.5 Спектральний аналіз сигналу	12
1.6 Розрахунок фільтра	13
1.7 Перегляд параметрів фільтра	16
1.8 Фільтрування сигналу	17
1.9 Збереження результатів роботи	18
2 ОСНОВНІ ФУНКЦІЇ ПАКЕТУ SP TOOLBOX	20
2.1 Функції роботи з аналоговими лінійними системами	21
2.2 Функції аналізу дискретних лінійних систем	25
2.3 Функції дискретної фільтрації	28
2.4 Функції синтезу дискретних фільтрів	
2.5 Функції спектрального аналізу і статистичної обробки сигналів	
2.6 Функції параметричного моделювання	
2.7 Допоміжні функції	
3 ОТРИМАННЯ ДАНИХ З ЗОВНІШНІХ ДЖЕРЕЛ	
3.1 Читання wav-файлів	
3.2 Запис wav-файлів	
3.3 Відтворення звуку	
4 ПІДВИЩЕННЯ ШВИДКОСТІ І ТОЧНОСТІ РОЗРАХУНКІВ	
4.1 Підвищення швидкості розрахунку	
4.2 Підвищення точності розрахунку	
ВИСНОВКИ	59
ПЕРЕЛІК ПОСИЛАНЬ	60

ВСТУП

Забезпечення надійної роботи різних машин неможливо без широкого використання на всіх стадіях їх життєвого циклу технічних засобів діагностування. Аварійні ситуації на АЕС, ТЕС, нафто - і газопроводах можуть бути значно зменшені або зовсім виключені, якщо ефективно застосовувати автоматизовану діагностичну техніку.

Віброакустична діагностика є найбільш прийнятною для оцінки технічного стану роторних машин та механізмів. Вона найбільш повно відображає природу явищ, робочих процесів, що протікають в них і використовує як джерело інформації вібросигнал, який має велику інформаційну ємність, малу інерційність, велику швидкість поширення, широкий частотний і динамічний діапазони. Ці якості вібросигналу дозволяють оперативно, а при використанні ЕОМ і в реальному масштабі часу, оцінювати технічний стан об'єкта діагностування.

Для реалізації методів вібродіагностики використовуються різні програмні комплекси такі як: MATLAB, LabVIEW, ZetLab, Spectrlab, MathCad ... Вони дозволяють повною мірою досліджувати вібросигнал використовуючи різні методи діагностування. Програмні комплекси замінили громіздке обладнання, яке потрібно було для збору та обробки вібросигналів, для проведення його аналізу та безпосередньо діагностування. Тепер для повної обробки вібросигналу та отримання інформації про стан машини потрібні фактично лише датчик і ПК з відповідним програмним забезпеченням.

MATLAB – одна з найстаріших, ретельно опрацьованих та перевірених часом систем автоматизації математичних та науково-технічних розрахунків, побудована на розширеному представленні та застосуванні матричних операцій. Це відбилося у назві системи – MATrix LABoratory – матрична лабораторія. Застосування матриць як основних об'єктів системи сприяє різкому зменшенню числа циклів, які дуже поширені при виконанні

матричних обчислень звичайними мовами програмування високого рівня, і полегшення реалізації паралельних обчислень. Одним із основних завдань при створенні системи MATLAB завжди було надання користувачам потужної мови програмування, орієнтованої на технічні та математичні розрахунки та здатного перевершити можливості традиційних мов програмування, які багато років використовувалися для реалізації чисельних методів. При цьому особлива увага приділялася як підвищенню швидкості обчислень, так і адаптації системи до вирішення найрізноманітніших завдань користувачів.

Метою даної роботи є дослідження можливостей системи комп'ютерного моделювання MATLAB для обробки і аналізу діагностичної інформації про технічний стан роторних машин.

Об'єкт дослідження: динамічні процеси у роторних машинах.

Предмет дослідження: можливості системи комп'ютерного моделювання MATLAB для обробки і аналізу діагностичної інформації про технічний стан роторних машин.

Для досягнення поставленої мети в роботі поставлено та вирішено такі основні задачі:

- показати, що віброакустична діагностика є найбільш прийнятною для оцінки технічного стану роторних машин та механізмів, як така, що найбільш повно відображає природу явищ та робочих процесів, що протікають в них;

- показати, що система MATLAB є зручним програмним засобом для створення систем збору та обробки вимірювальної інформації на основі технології віртуальних приладів і дозволяє розробляти діагностичні системи практично будь-якої складності;

- показати, що система MATLAB є зручною прикладною системою програмування і моделювання, яку ефективно можна використовувати для математичного моделювання механічних систем та дослідження їх динамічних характеристик.

1 ВИКОРИСТАННЯ СИСТЕМИ КОМП'ЮТЕРНОГО МОДЕЛЮВАННЯ MATLAB ДЛЯ ОБРОБКИ І АНАЛІЗУ ДІАГНОСТИЧНОЇ ІНФОРМАЦІЇ

1.1 Система MATLAB і її можливості по обробці сигналів

До складу системи MATLAB входить Signal Processing Toolbox (SPTool) - потужний пакет з аналізу, моделювання та проектування пристроїв обробки всіляких сигналів, забезпечення їх фільтрації та безлічі перетворень.

Пакет Signal Processing забезпечує надзвичайно великі можливості створення програм обробки сигналів для сучасних наукових та технічних програм. У пакеті використовується різноманітна техніка фільтрації та новітні алгоритми спектрального аналізу. Пакет містить модулі для розробки лінійних систем та аналізу часових рядів. Пакет може використовуватися, зокрема, в таких галузях, як обробка аудіо- та відеоінформації, телекомунікації, геофізика, завдання управління в реальному режимі часу, економіка, фінанси та медицина. Основні властивості пакета:

- моделювання сигналів та лінійних систем;
- проектування, аналіз та реалізація цифрових та аналогових фільтрів;
- швидке перетворення Фур'є, дискретне косинусне та інші перетворення;
- оцінка спектрів та статистична обробка сигналів;
- параметричне оброблення часових рядів;
- генерація сигналів різної форми.

Пакет Signal Processing – ідеальна оболонка для аналізу та обробки сигналів. У ньому використовуються перевірені практикою алгоритми, обрані за критеріями максимальної ефективності та надійності. Пакет містить широкий спектр алгоритмів для представлення сигналів та лінійних моделей. Цей набір дозволяє користувачеві досить гнучко підходити до створення сценарію обробки сигналів. Пакет включає алгоритми для перетворення моделі з одного уявлення в інше.

Пакет Signal Processing включає повний набір методів створення цифрових фільтрів з різноманітними характеристиками. Він дозволяє швидко розробляти фільтри верхніх і нижніх частот, смугові фільтри, що пропускають і затримують, багатосмугові фільтри, у тому числі фільтри Чебишева, Юла-Уолкера, еліптичні та ін. Графічний інтерфейс дозволяє проектувати фільтри, задаючи вимоги до них як перенесення об'єктів мишею.

Заснований на оптимальному алгоритмі швидкого перетворення Фур'є пакет Signal Processing має неперевершені характеристики для частотного аналізу та спектральних оцінок. Пакет включає функції для обчислення дискретного перетворення Фур'є, дискретного косинусного перетворення, перетворення Гільберта та інших перетворень, які часто застосовуються для аналізу, кодування та фільтрації. У пакеті реалізовані такі методи спектрального аналізу, як метод Вельха, метод максимальної ентропії та ін.

Новий графічний інтерфейс дозволяє переглядати та візуально оцінювати характеристики сигналів, проектувати та застосовувати фільтри, проводити спектральний аналіз, досліджуючи вплив різних методів та їх параметрів на отриманий результат. Графічний інтерфейс особливо корисний для візуалізації часових рядів, спектрів, часових та частотних характеристик, розташування нулів та полюсів передавальних функцій систем.

Пакет Signal Processing є основою вирішення багатьох інших завдань. Наприклад, комбінуючи його з пакетом Image Processing, можна обробляти та аналізувати двовимірні сигнали та зображення. У парі з пакетом System Identification пакет Signal Processing дозволяє виконувати параметричне моделювання систем у часовій області. У поєднанні з пакетами Neural Network та Fuzzy Logic може бути створено багато засобів для обробки даних або виділення класифікаційних характеристик. Засіб генерації сигналів дозволяє створювати імпульсні сигнали різної форми.

Таким чином, програма Signal Processing Tool надає найширші можливості користувачу з обробки сигналів, зокрема для вібродіагностики технічного стану машин та механізмів.

1.2 Програма Signal Processing Tool (SPTool): головне вікно

Програма SPTool надає у розпорядження користувача графічне середовище для перегляду графіків сигналів та їх спектрів, розрахунку та аналізу фільтрів, а також фільтрації сигналів. Для запуску програми SPTool необхідно набрати її ім'я у командному рядку MAT LAB: `sptool`. Після цього з'явиться основне вікно програми, показане на рисунку 1.1.

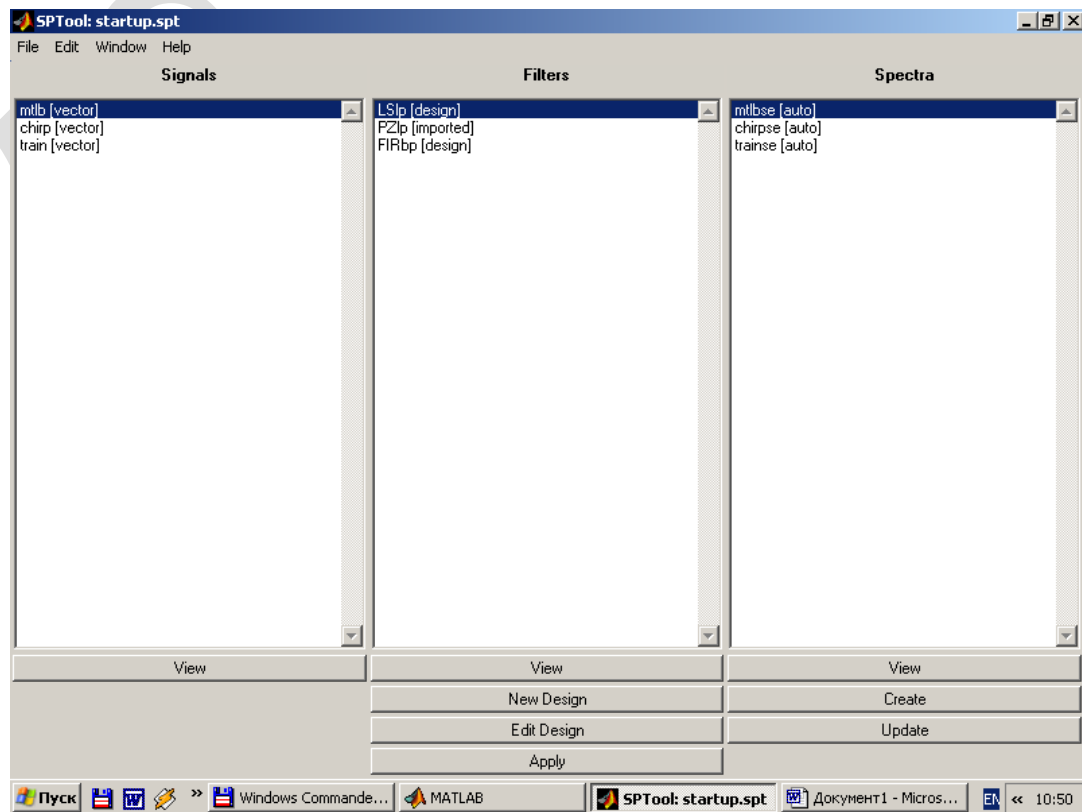


Рисунок 1.1 – Програма SPTool (головне вікно)

У трьох списках цього вікна перераховані завантажені в програму SPTool сигнали (Signals), фільтри (Filters) та спектри (Spectra). Розміщені під списками кнопки дозволяють виконувати різні операції, більшість з яких ми розглянемо далі.

Типовий набір дій, що виконуються за допомогою програми SPTool, включає наступні операції: завантаження сигналу; перегляд графіка сигналу; спектральний аналіз сигналу; розрахунок фільтра; фільтрація сигналу.

Далі коротко розглянемо виконання всіх перерахованих процедур.

1.3 Завантаження сигналу

Для завантаження сигналу в меню File головного вікна SPTool необхідно вибрати команду Import. З'явиться вікно Import to SPTool, показане на рисунку 1.2.

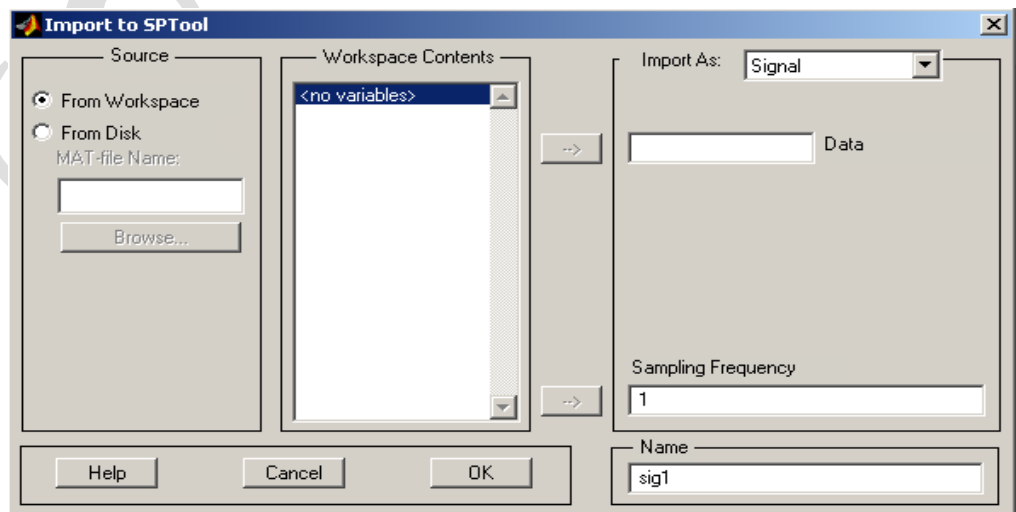


Рисунок 1.2 - Програма SPTool (вікно імпорту даних)

Перемикач Source дозволяє вказати місце зберігання сигналу, що завантажується - робочу область MATLAB (From Workspace) або MAT-файл (From Disk). У другому випадку стає доступним поле введення MAT-file Name, яке потрібно ввести вручну або за допомогою кнопки Browse ім'я потрібного MAT-файлу.

У списку Workspace Contents перераховані змінні, що є у робочій пам'яті MATLAB на даний момент. Якщо в якості джерела було обрано MAT-файл, цей список буде називатися File Contents і перераховувати змінні, збережені у вибраному файлі.

У списку Import As, що розкривається, потрібно вибрати варіант Signal. Інші два варіанти, Filter і Spectrum, дозволяють імпортувати описи фільтрів

та вже розрахованих спектрів для перегляду та аналізу.

Потім потрібно вибрати в списку змінну, що містить відліки сигналу, що завантажується, і клацнути на кнопці -->, розташованій поруч з полем введення Data. Також можна вручну ввести в це поле ідентифікатор змінної.

У полі введення Sampling Frequency за замовчуванням знаходиться значення 1. Його можна відредагувати вручну, а можна імпортувати, ввівши ідентифікатор змінної або вибравши його у списку та клацнувши по нижній кнопці -->.

Нарешті, у полі введення Name можна відредагувати ім'я, під яким цей сигнал фігуруватиме у програмі SPTool. Зробивши це, натисніть кнопку ОК. Імпортований сигнал з'явиться у списку Signals основного вікна програми (рис.1.1).

1.4 Перегляд графіка сигналу

Для перегляду графіка сигналу потрібно вибрати його у списку Signals (рисунок 1.1) та клацнути на кнопці View, розташованій під цим списком. З'явиться вікно Signal Browser, показане на рисунку 1.3.

У цьому вікні виводяться два графіки. Нижній з написом Panner показує панораму всього сигналу. Верхній графік спочатку теж відображає весь сигнал, але масштаб відображення можна збільшити, і тоді верхній графік показуватиме лише фрагмент сигналу, а положення цього фрагмента загальної панорами демонструється за допомогою прямокутника на нижньому графіку. Цей прямокутник можна перетягувати мишею, змінюючи тим самим ділянку огляду.

Кнопки панелі інструментів вікна Signal Browser дозволяють роздруковувати графік сигналу, відтворювати сигнал за допомогою звукової карти комп'ютера, керувати масштабом відображення, вибирати потрібний канал у разі багатоканального сигналу та задавати потрібний режим відображення маркерів.

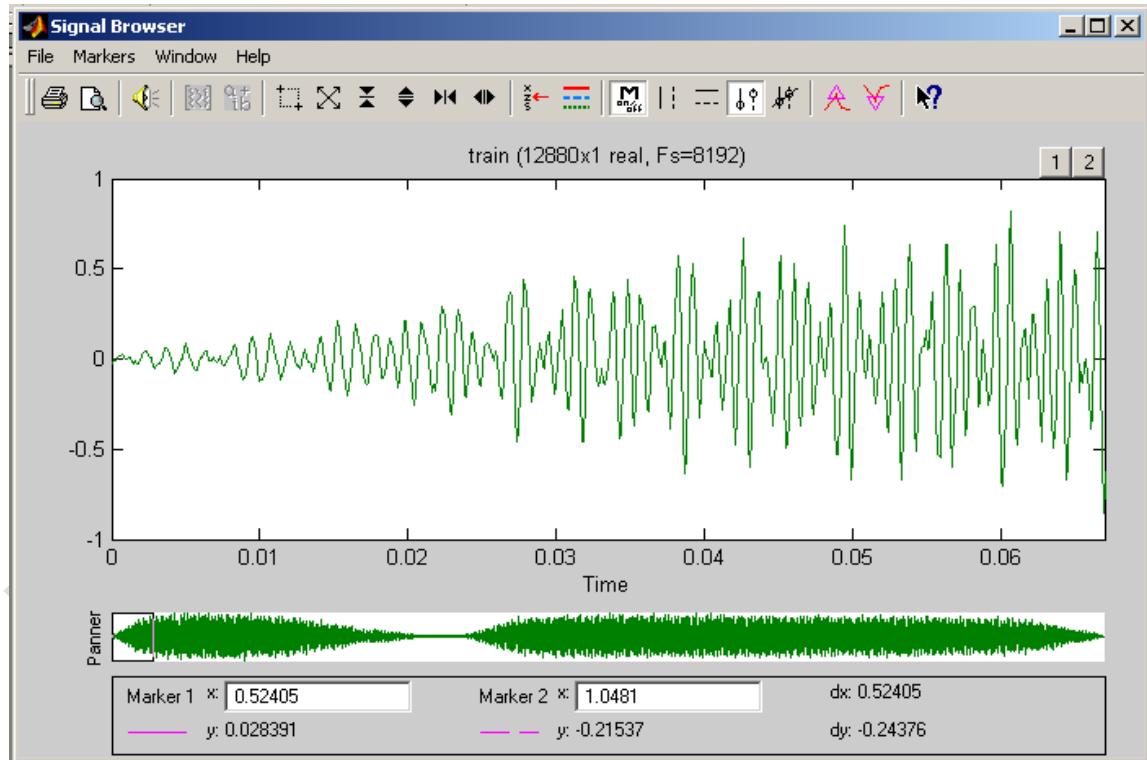


Рисунок 1.3 - Програма SPTool (вікно перегляду сигналів)

На графік можна нанести два маркери, що дозволяють робити над сигналом кількісні виміри. Маркери перетягуються за допомогою миші, а інформація про помічені ними відліки сигналу виводиться в нижній частині вікна. Можна експортувати цю інформацію до робочої області пам'яті MATLAB у вигляді структури - для цього є команда Export меню Markers.

1.5 Спектральний аналіз сигналу

Для аналізу діапазону сигналу, завантаженого в програму SPTool, необхідно вибрати потрібний сигнал у списку Signals головного вікна програми (рис. 1.1) і клацнути на кнопці Create, розташованій під списком Spectra. З'явиться вікно Spectrum Viewer, показане на рисунку 1.4.

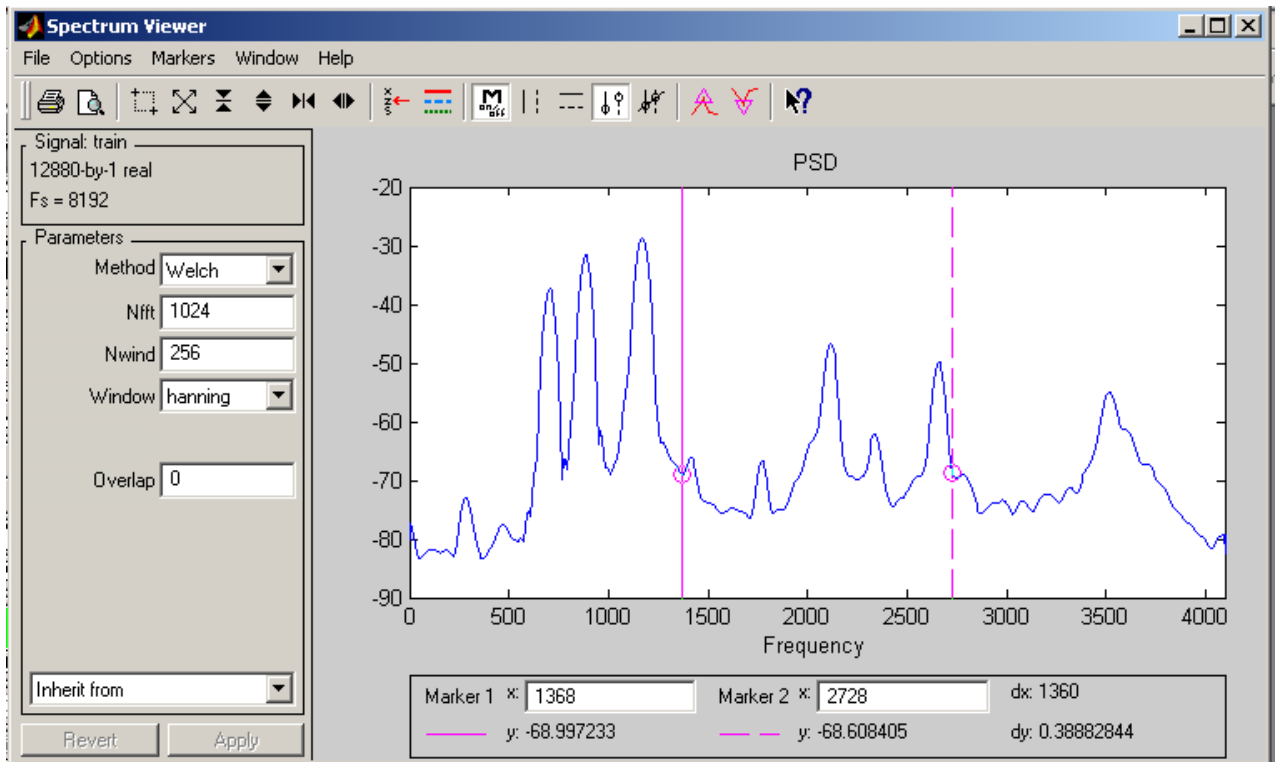


Рисунок 1.4 - Програма SPTool (вікно спектрального аналізу)

У лівій частині вікна вибирається метод спектрального аналізу та налаштовуються його параметри. Список, що розкривається, в якому спочатку виведений рядок Inherit from, дозволяє скопіювати повний набір налаштувань аналізатора спектра з іншого розрахунку, представленого в списку Spectra основного вікна програми.

Здійснивши налаштування параметрів аналізу, клацніть на кнопці Apply. Буде розраховано оцінку спектра сигналу та виведено відповідний графік. Кнопки панелі інструментів дозволяють виконувати ті самі операції, що були перераховані вище стосовно перегляду графіків сигналів. При перегляді спектра також можливе використання маркерів.

1.6 Розрахунок фільтра

Для розрахунку дискретного фільтра потрібно клацнути на кнопці New Design, розташованій під списком Filters у головному вікні SPTool (рисунок

1.1). Можна також змінити параметри вже розрахованого фільтра, обравши його та списку Filters та клацнувши на кнопці Edit Design. У кожному з наведених випадків з'явиться вікно Filter Designer, показане на рисунку 1.5.

У списку Algorithm, що розкривається, вибирається метод розрахунку, а в розділі Specifications задаються параметри синтезованого фільтра.

Вибравши метод розрахунку та задавши параметри фільтра, потрібно клацнути на кнопці Apply для виконання синтезу. У центрі вікна буде виведений графік АЧХ фільтра, а в розділі Measurements показані його параметри.

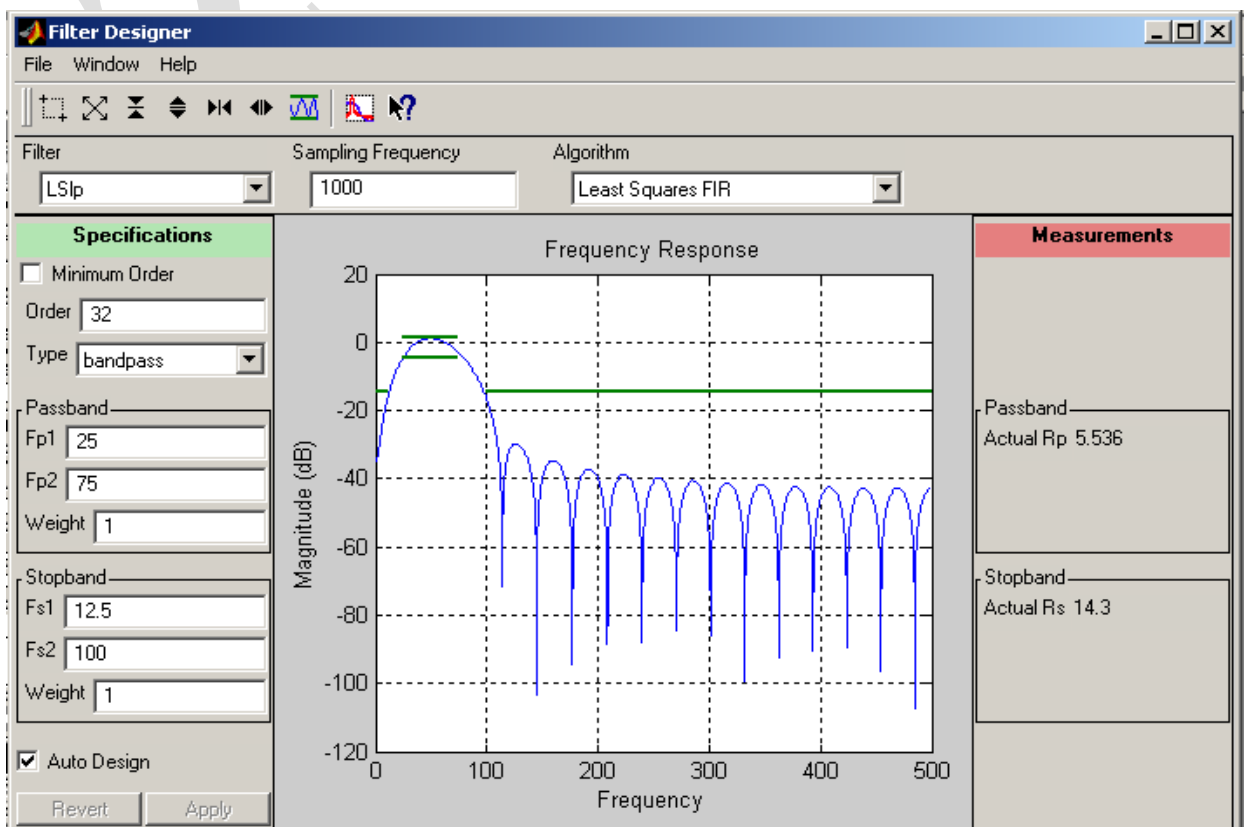


Рисунок 1.5 - Програма SPTool (вікно розрахунку фільтрів)

Кнопки панелі інструментів дозволяють керувати масштабом графіки. Крім того, можна вивести графік АЧХ у смузі пропускання крупним планом, а також накласти на АЧХ графік одного із спектрів, перерахованих у списку Spectra основного вікна програми.

Можливості програми SPTool із синтезу фільтрів суттєво обмежені порівняно з програмою FDATool. Єдина можливість, що є у SPTool і відсутня у FDATool, - це пряме редагування розташування нулів та полюсів фільтра. Для такого редагування виберіть у списку Algorithm останній рядок – Pole/Zero Editor. Вікно Filter Designer набуде вигляду, показаного на рисунку 1.6.

У цьому режимі редагування фільтра можна переміщати нулі та полюси мишею, додавати і видаляти їх, автоматично формувати їх комплексно-сполучені пари.

Оскільки у вікні Filter Designer при редагуванні нулів і полюсів не виводиться графік АЧХ фільтра, при використанні даного режиму слід одночасно з вікном Filter Designer відкрити вікно Filter Viewer, що розглядається далі. При редагуванні фільтра у вікні Filter Designer його характеристики, що відображаються у вікні Filter Viewer, будуть синхронно оновлюватися. Взагалі, вікна програми SPTool не є модальними, тобто можна відкрити будь-яку їх кількість і вільно перемикатися між ними.

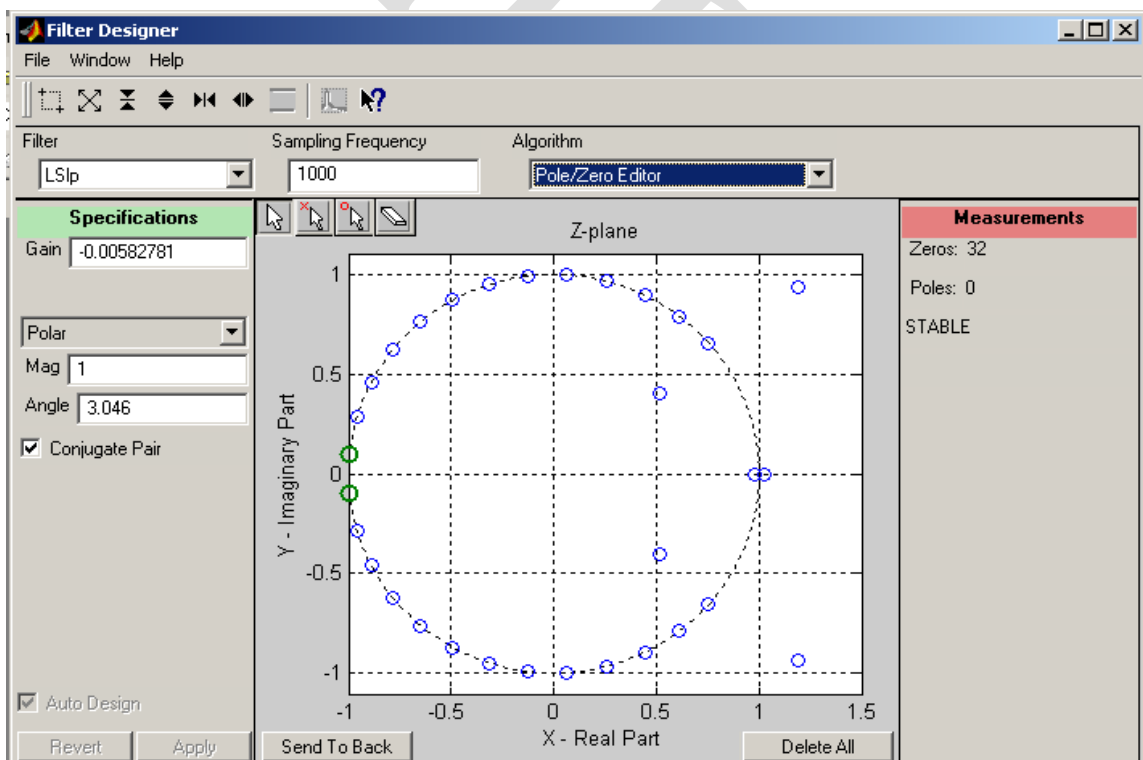


Рисунок 1.6 - Програма SPTool
(редагування розміщення нулів і полюсів фільтра)

1.7 Перегляд параметрів фільтра

Для перегляду параметрів фільтра, завантаженого в програму SPTool, необхідно вибрати його у переліку Filters основного вікна програми (рис. 1.1) і клацнути на кнопці View, розташованій під цим переліком. З'явиться вікно Filter Viewer, показане малюнку 1.7.

У лівій частині вікна Filter Viewer розташована група прапорців Plots для вибору складу відображуваних графіків. У розділі Frequency Axis можна вибрати частотний діапазон для перегляду характеристик і встановити тип шкали частот - лінійний або логарифмічний.

Кнопки панелі інструментів дозволяють керувати режимом відображення та виведенням маркерів для поточного графіка, осі якого виділяються червоним кольором. Інформація про маркери, як завжди, виводиться в нижній частині вікна.

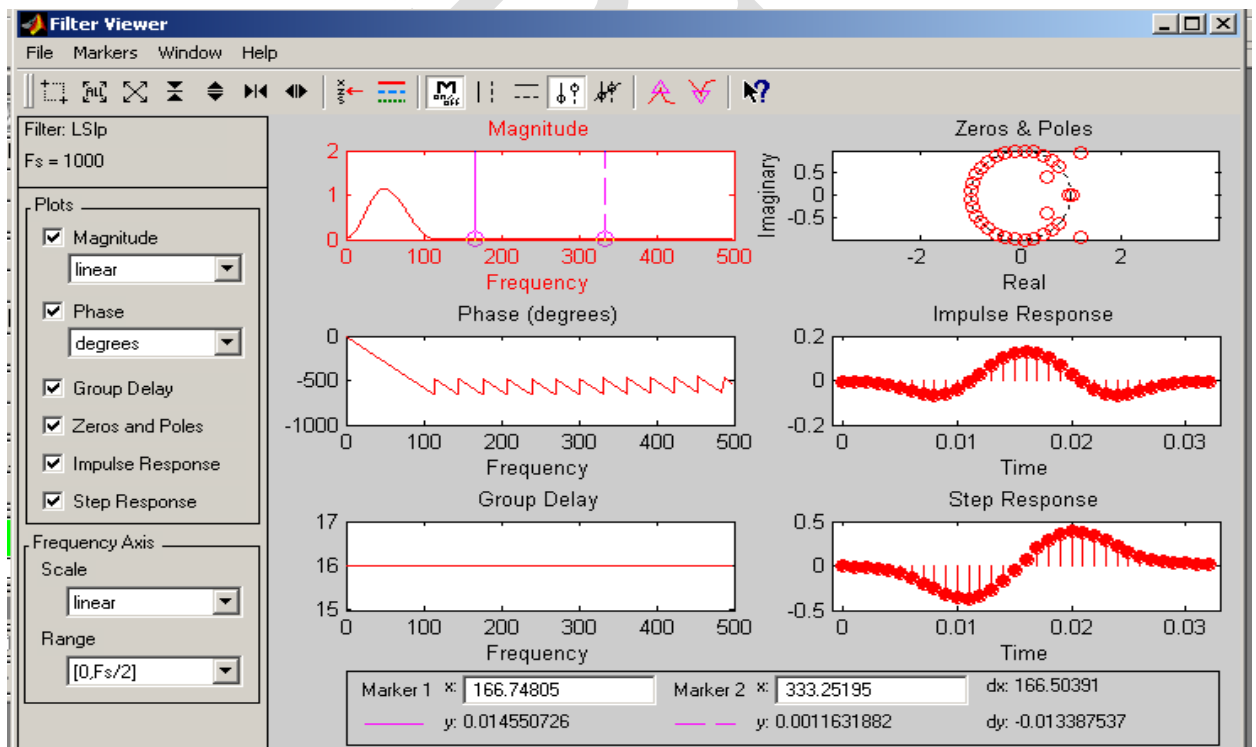


Рисунок 1.7 - Програма SPTool (перегляд характеристик фільтра)

1.8 Фільтрування сигналу

Для пропускання сигналу через фільтр необхідно вибрати сигнал і фільтр відповідно у списках Signals та Filters основного вікна програми (малюнок 1.1), а потім натиснути на кнопці Apply, розташованій під списком Filters. З'явиться вікно Apply Filter, показане на рисунку 1.8.

Єдиним доступним полем введення цього вікна є поле Output Signal, у якому потрібно задати ім'я для вихідного сигналу. У списку Algorithm, що розкривається, можна вибрати функцію MATLAB для здійснення фільтрації - filter, filtfilt або fftfilt. Виконавши ці дії, потрібно клацнути на кнопці ОК. Буде розрахований вихідний сигнал, який з'явиться під вказаним ім'ям у списку Signals основного вікна програми.

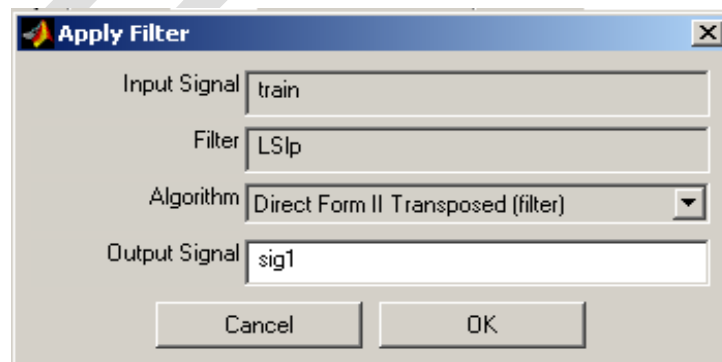


Рисунок 1.8 - Програма SPTool (вікно застосування фільтра)

Переглянути графік вихідного сигналу і виконати аналіз його спектра можна описаними способами. Для одночасного перегляду графіків вхідного та вихідного сигналів потрібно вибрати їх обидва у списку Signals основного вікна програми (для цього при клацанні на сигналі необхідно, як це прийнято в Windows, натиснути клавішу Ctrl) і клацнути на кнопці View. У вікні Signal Browser, що відкриється, будуть показані графіки обох вибраних сигналів (рисунок 1.9)

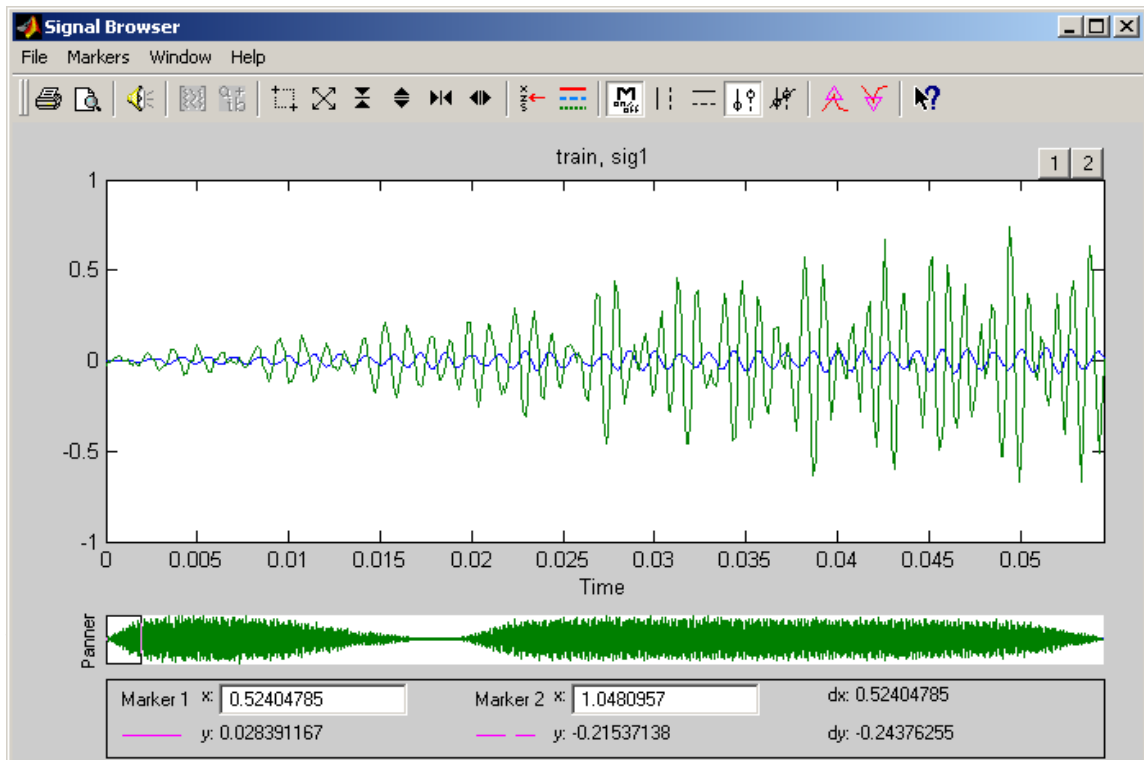


Рисунок 1.9 - Програма SPTool (одночасний перегляд графіків вхідного і вихідного сигналів фільтра)

Аналогічно можна переглядати кілька спектрів або характеристики декількох фільтрів одночасно.

1.9 Збереження результатів роботи

Сеанс роботи з SPTool можна зберегти за допомогою команди Save Session або Save Session As з меню File основного вікна програми. Файли сеансів мають розширення SPT. Завантажити збережений сеанс можна командою Open Session того ж меню File.

Крім того, можна експортувати сигнали, фільтри та спектри у вигляді структур даних. Для цього використовується команда Export із меню File основного вікна програми SPTool. Після вибору цієї команди з'явиться вікно Export from SPTool, показане на рисунку 1.10.

У списку, що займає ліву частину цього вікна, необхідно вибрати

об'єкти, що експортуються. Клацніть на кнопці Export to Workspace експортувати відповідні структури даних у робочу область пам'яті MATLAB, а для їх запису в MAT-файл необхідно скористатися кнопкою Export to Disk.

Інформацію про склад експортованих структур даних можна отримати із довідкової системи пакету Signal Processing.

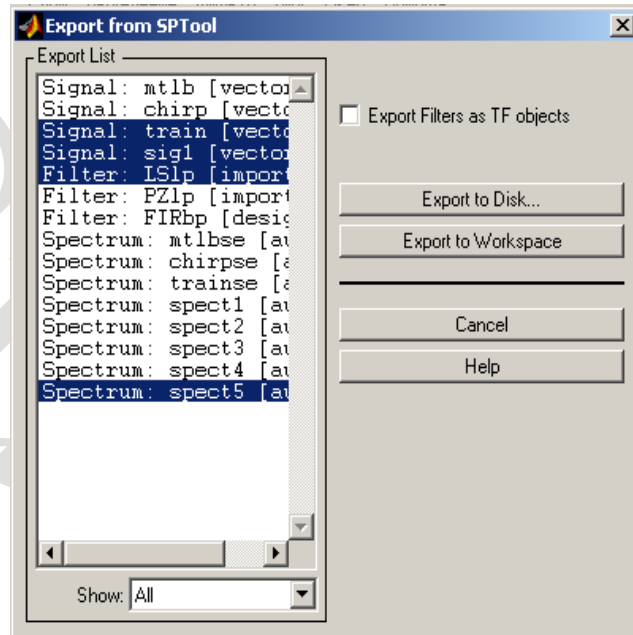


Рисунок 1.10 - Програма SPTool (вікно експорту даних)

2 ОСНОВНІ ФУНКЦІЇ ПАКЕТУ SP TOOLBOX

Пакет Signal Processing містить майже двісті ретельно розроблених спеціалізованих функцій, що дозволяють вирішувати найрізноманітніші завдання аналізу та обробки сигналів.

Відповідно до офіційної документації пакета його функції поділяються на двадцять категорій. У наведеному нижче списку ці категорії об'єднані в більші групи. Отже, за призначенням функції пакета Signal Processing можна розділити так:

- функції роботи з аналоговими лінійними системами;
- функції аналізу дискретних лінійних систем;
- функції дискретної фільтрації;
- функції синтезу дискретних фільтрів;
- функції спектрального аналізу та статистичної обробки сигналів;
- функції параметричного моделювання та лінійного передбачення;
- функції генерації сигналів;
- функції перетворень дискретних сигналів;
- допоміжні функції.

Крім того, до складу пакету входять три графічні середовища:

- програма синтезу та аналізу фільтрів – FDATool (Filter Design & Analysis Tool);
- програма обробки сигналів - SPTool (Signal Processing Toolbox)
- програма синтезу та аналізу вагових (віконних) функцій – WinTool (Windows Design & Analysis Tool).

2.1 Функції роботи з аналоговими лінійними системами

MATLAB та його пакети розширення орієнтовані насамперед на цифрову обробку сигналів, тому функції, пов'язані з розрахунком аналогових ланцюгів, розглядаються як допоміжні. В основному, вони призначені для

виклику з інших функцій, що використовують аналогові прототипи при синтезі цифрових фільтрів. Однак і самі по собі ці функції можуть бути дуже корисними. У свою чергу їх можна розділити на кілька груп.

Перша група - функції розрахунку аналогових фільтрів-прототипів, тобто фільтрів нижніх частот із частотою зрізу, що дорівнює 1 рад/с. Функції повертають нулі, полюси та коефіцієнт посилення фільтрів і дозволяють розраховувати фільтри-прототипи Баттерворта (**buttap**), Чебишева першого роду (**cheb1ap**), Чебишева другого роду (**cheb2ap**), еліптичні (Кауера) (**ellipap**) та Бесселя (**besselap**).

Друга група — функції перетворення аналогових фільтрів, що дозволяють перетворити ФНЧ-прототип у ФНЧ з іншою частотою зрізу (**lp2lp**), у фільтр верхніх частот (ФВЧ) із заданою частотою зрізу (**lp2hp**), у смуговий фільтр із заданими середньою частотою та шириною смуги пропускання (**lp2bp**) та в режекторний фільтр із заданими середньою частотою та шириною смуги затримувannya (**lp2bs**).

Третя група – функції розрахунку аналогових фільтрів із заданими параметрами. У процесі розрахунку вони викликають функції перших двох груп. Набір необхідних розрахунку вихідних даних включає порядок фільтра, його тип (ФНЧ, ФВЧ, полосовий чи режекторний), частоту чи кілька частот зрізу, і навіть (залежно від прототипу) параметри пульсацій амплітудно-частотної характеристики (АЧХ). Є функції для розрахунку фільтрів Баттерворта (**butter**), Чебишева першого роду (**cheby1**), Чебишева другого роду (**cheby2**), еліптичних (Кауера) (**ellip**) та Бесселя (**besself**). Всі ці функції, крім функції **besself**, можуть використовуватися для розрахунку дискретних фільтрів. Ознакою аналогового варіанта розрахунку є вказівка рядка «s» як останнього вхідного параметра.

Четверта група - функції визначення необхідного порядку фільтра за заданими параметрами АЧХ (граничних частот смуг пропускання та затримувannya, а також допустимих пульсацій у смугах пропускання та затримувannya). Для кожного типу фільтра є своя функція визначення

необхідного порядку: для фільтра Баттерворта – **buttord**, для фільтра Чебишева першого роду – **cheb1ord**, для фільтра Чебишева другого роду – **cheb2ord**, для еліптичного фільтра – **ellipord**. Так само, як і функції попередньої групи, ці функції дозволяють визначати необхідний порядок і для дискретних фільтрів (див. далі). Ознакою аналогового варіанта розрахунку є вказівка рядка «s» як останнього вхідного параметра.

П'ята група – функції перетворення форм опису аналогових лінійних систем. Для аналогових систем підтримуються чотири такі способи опису:

- коефіцієнти поліномів чисельника та знаменника функції передачі системи;
- нулі, полюси та коефіцієнт посилення системи (розкладання функції передачі на множники);
- полюси та відрахування (подання функції передачі системи у вигляді суми простих дробів);
- параметри простору станів.

У пакеті Signal Processing є функції, що реалізують взаємні перетворення цих чотирьох форм представлення аналогових систем (тільки функція `residue`, що працює з полюсами та відрахуваннями, відноситься не до пакету Signal Processing, а до базової бібліотеки MATLAB).

Нарешті, до шостої групи слід віднести єдину функцію **freqs**, що дозволяє розрахувати або відобразити графічно амплітудно- та фазочастотну характеристики (АЧХ та ФЧХ) аналогової лінійної системи. Вихідними даними є коефіцієнти поліномів чисельника та знаменника функції передачі системи.

Як приклад застосування функцій роботи з аналоговими системами розрахуємо еліптичний ФНЧ четвертого порядку з частотою зрізу 3 кГц, пульсаціями АЧХ у смузі пропускання, рівними 1 дБ, та з придушенням сигналу у смузі затримування, рівним 20 дБ, а потім побудуємо графіки його А (Рисунок 2.1).

```
[b, a] = ellip (4, 1, 20, 2 * pi * 3000, 's'); % розрахунок фільтра
```

```
f=0:10:10000; % вектор частот для розрахунку АЧХ та ФЧХ
h = freqs (b, a, 2 * pi * f); % комплексний коефіцієнт передачі
subplot(2, 1, 1)
plot(f, abs(h)) % графік АЧХ
grid
subplot(2, 1, 2)plot(f, unwrap(angle(h))*180/pi) % графік ФЧХ
grid
```

Використана в прикладі коду функція **unwrap** усуває незначні стрибки ФЧХ на 360° .

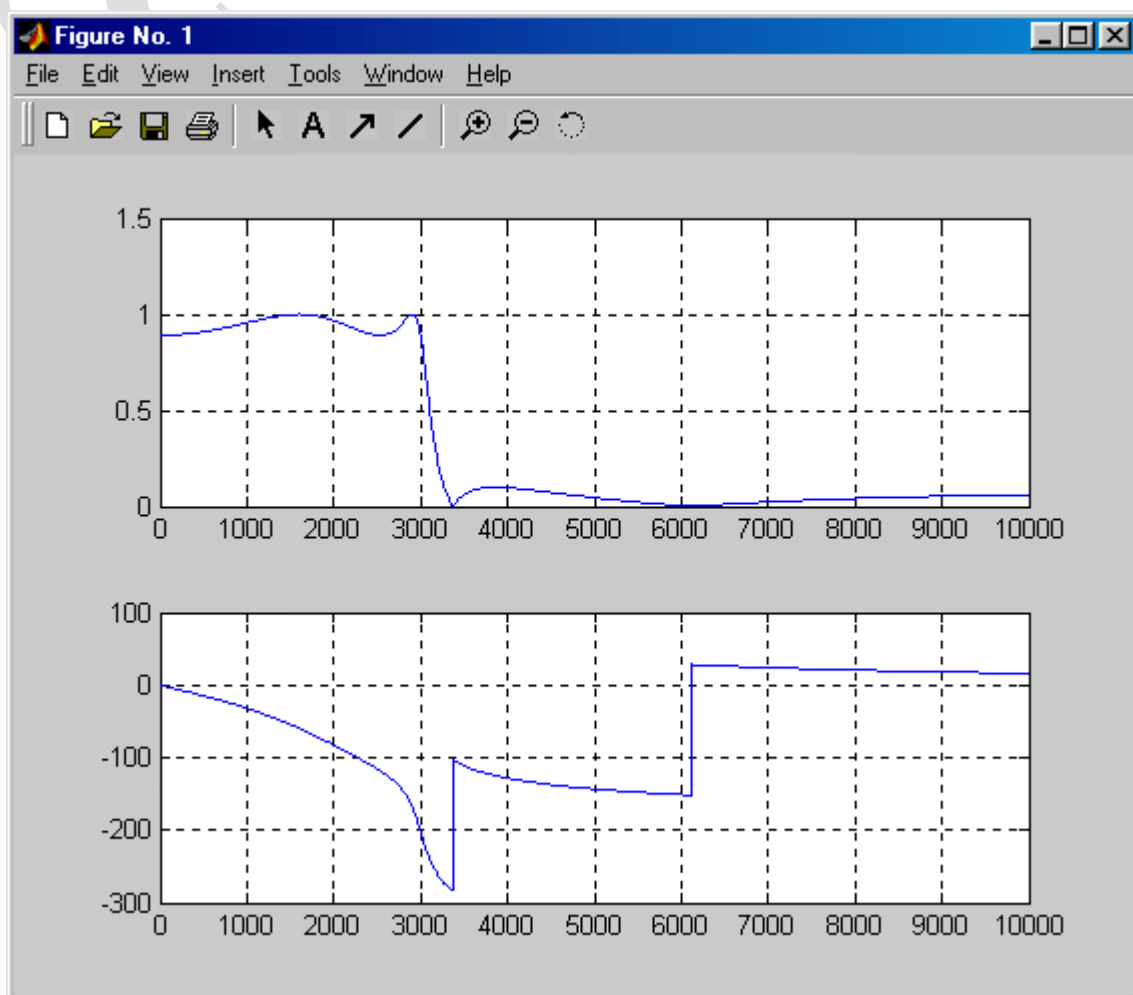


Рисунок 2.1 – Графіки АЧХ і ФЧХ

2.2 Функції аналізу дискретних лінійних систем

Ця категорія функцій є досить численною і поєднує різноманітні засоби аналізу дискретних лінійних систем — як правило, представлених у вигляді векторів коефіцієнтів поліномів чисельника та знаменника функції передачі (z -області).

Функція **freqz** є дискретним аналогом функції **freqs**, вона дозволяє розрахувати комплексний коефіцієнт передачі чи побудувати графіки АЧХ і ФЧХ дискретної системи. Вихідними даними є коефіцієнти поліномів чисельника та знаменника функції передачі системи.

Аналогічно функції **freqz** працює функція **grpdelay**, що дозволяє розрахувати або графічно відобразити частотну залежність групової затримки дискретним фільтром.

Функція **impz** призначена для розрахунку чи графічного відображення імпульсної характеристики дискретної системи. Вихідними даними, як і попередніх функцій, є коефіцієнти поліномів чисельника і знаменника функції передачі системи.

Функція **zplane** дозволяє відобразити нулі та полюси системи на комплексній площині, додатково зобразивши одиничне коло, що обмежує допустиму область розташування полюсів сталої дискретної системи. Вихідними даними можуть бути як вектори коефіцієнтів поліномів чисельника та знаменника, так і безпосередньо вектори нулів та полюсів функції передачі системи.

Функція **filternorm** дозволяє розрахувати норму дискретного фільтра. Цей параметр використовується, наприклад, при виборі коефіцієнтів масштабування окремих секцій фільтра, реалізованого в послідовній (каскадній) формі, для зменшення помилок округлення. Підтримуються два варіанти: 2-норма та ∞ -норма. 2-норма є середньоквадратичним значенням АЧХ фільтра (або, що те саме, корінь із суми квадратів відліків імпульсної характеристики фільтра), а ∞ -норма — це максимальне значення АЧХ.

Функція **fvtool** по суті є графічним середовищем, призначеним для аналізу та візуалізації характеристик дискретних систем (Filter Visualization Tool). Однак, на відміну від інших графічних середовищ пакета, **fvtool** дійсно є функцією, оскільки при виклику вимагає наявності вхідних параметрів - коефіцієнтів поліномів чисельника та знаменника функції передачі фільтра, що аналізується. Істотною перевагою даної функції є можливість одночасного перегляду характеристик кількох фільтрів. Графічний інтерфейс користувача, що забезпечується цією функцією, практично такий самий, як у середовищі аналізу та синтезу фільтрів **FDATool**. Нижче показаний приклад виклику функції **fvtool** і створюване їй графічне вікно (рисунок 2.2) у режимі показу частотної залежності, що вноситься фільтром групової затримки.

```
b = [121];
a = [1 -1.1 0.9];
fvtool(b, a)
```

Велику групу складають функції перетворення форм описів дискретних систем. Для дискретних систем підтримується більше форм уявлення, ніж аналогових:

- коефіцієнти поліномів чисельника та знаменника функції передачі системи;
- нулі, полюси та коефіцієнт посилення системи (розкладання функції передачі на множники);
- полюси та відрахування (подання функції передачі системи у вигляді суми простих дробів);
- параметри простору станів;
- подання у вигляді набору послідовно (каскадно) включених секцій другого порядку;
- подання у вигляді гратчастої або гратчасто-сходової структури.

Одна й та сама функція **residuez** (це аналог функції **residue**, призначений для роботи з описами дискретних систем) може здійснювати

перетворення в обох напрямках. Напрямок перетворення визначається кількістю вхідних та вихідних параметрів.

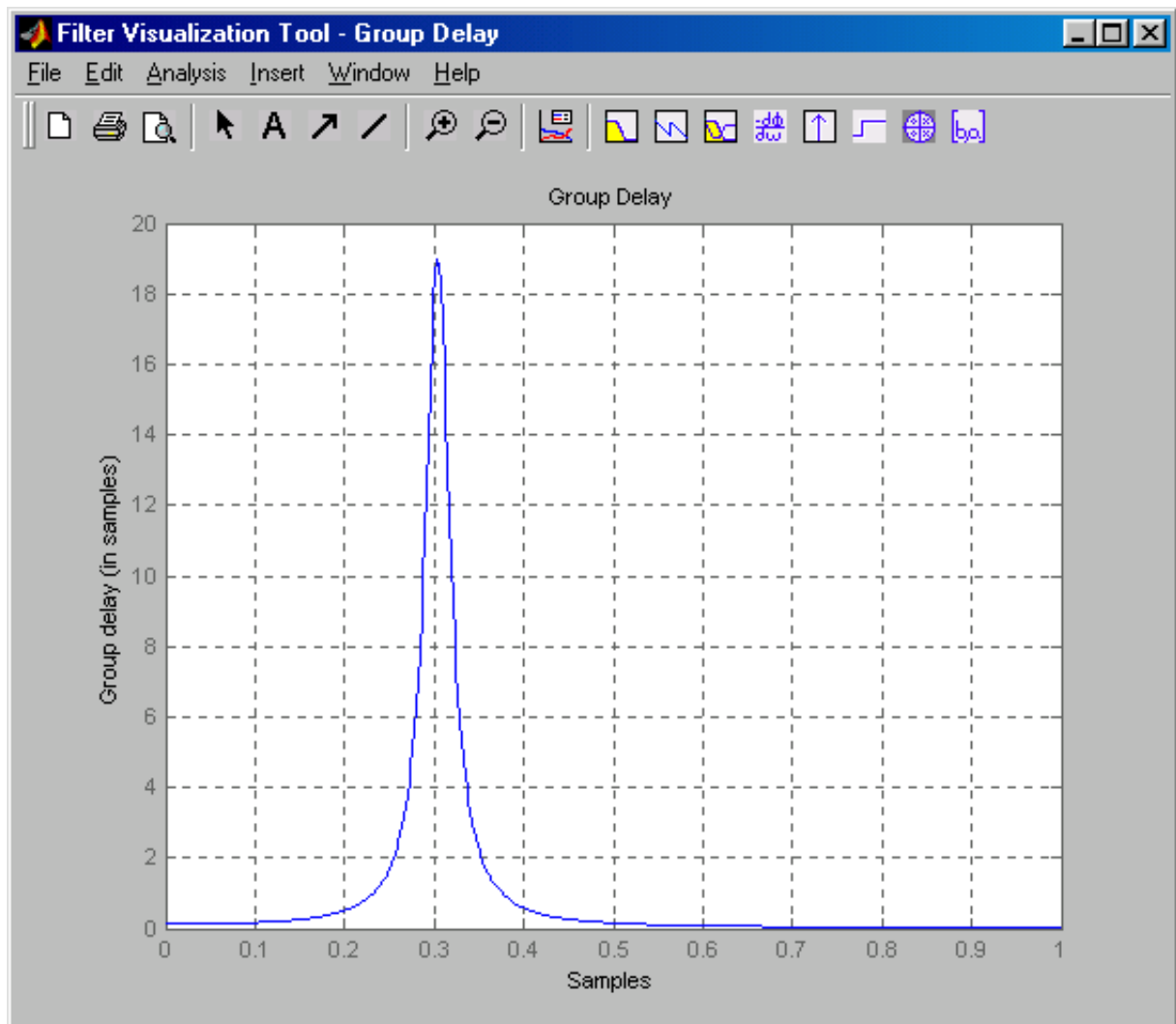


Рисунок 2.2 – Графічне вікно **fvtool**

Функція **freqspace** розраховує вектор рівномірно розташованих значень частот, функція **freqzplot** призначена для побудови графіків частотних характеристик, а функція **unwrap** дозволяє усунути незначні стрибки фазочастотних характеристик на 2π , відшуковуючи ці стрибки між елементами вектора і зсуваючи потрібні фрагменти вектора на 2π .

2.3 Функції дискретної фільтрації

Операція лінійної дискретної фільтрації в загальному випадку описується наступним чином:

$$y(k) = b_0x(k) + b_1x(k-1) + \dots + b_mx(k-m) - a_0y(k-1) - a_1y(k-2) - \dots - a_ny(k-n) \quad (2.1)$$

Тут $x(k)$ – відліки вхідного сигналу, $y(k)$ – відліки вихідного сигналу, a_i та b_j – постійні коефіцієнти. Максимальне число m і n називається порядком фільтра.

Попередні вихідні відліки можуть не використовуватися під час розрахунків, тоді всі $a_i = 0$ і фільтр називається нерекурсивним або трансверсальним. Якщо попередні вихідні відліки використовуються, фільтр називається рекурсивним.

Лінійна дискретна фільтрація відноситься до технологій обробки довільних даних, тому відповідна функція – **filter** – належить не пакету Signal Processing, а є вбудованою в ядро MATLAB. Функція **filter2**, що також належить базовій бібліотеці MATLAB, реалізує двовимірну дискретну фільтрацію.

Оскільки фільтр з постійними коефіцієнтами є лінійною стаціонарною дискретною системою, його реакція на довільний вхідний сигнал може бути представлена як дискретна згортка вхідного сигналу з імпульсною характеристикою фільтра:

$$y(k) = \sum_{n=-\infty}^{\infty} x(n)h(k-n) \quad (2.2)$$

Тут $h(k)$ – відліки імпульсної характеристики фільтра. Імпульсна характеристика є реакцією фільтра на поданий на вхід одиночний відлік одиничної величини.

Зрозуміло, обчислення за формулою згортки можуть бути реалізовані

практично лише за кінцевої довжині імпульсної характеристики фільтра. Ця операція здійснюється функцією **conv**; так само як і реалізація алгоритму дискретної фільтрації, вона відноситься не до пакету Signal Processing, а до базової бібліотеки MATLAB. Насправді, реалізація функції **conv** зводиться до виклику функції **filter** з відповідними вхідними параметрами. Функція **conv2** реалізує двовимірну дискретну згортку. Ще одна функція базової бібліотеки MATLAB – **deconv** – реалізує звернення згортки, дозволяючи за результатом згортки та одному з вхідних векторів визначити другий вхідний вектор.

Згадані вище базові функції дискретної фільтрації, як було зазначено, відносяться до базової бібліотеки MATLAB; власне ж у пакеті Signal Processing розташовані функції, що вирішують специфічні завдання фільтрації.

Насамперед, слід зазначити, що функція **filter** дає можливість доступу до початкового та кінцевого внутрішнім станам фільтра, дозволяючи за рахунок цього організувати блокову обробку сигналу. Іноді виникає необхідність сформуванню вектор внутрішнього стану фільтра, знаючи деяку кількість попередніх вхідних і вихідних відліків. Такий розрахунок виконується за допомогою функції **filtic**.

Функція **fftfilt** реалізує дискретну фільтрацію з допомогою швидкого перетворення Фур'є (БПФ) разом із поділом сигналу на блоки. У такий спосіб можуть бути реалізовані лише нерекурсивні фільтри. Результат роботи функції збігається (з точністю до обчислювальних похибок) з результатами звичайної фільтрації, що реалізується за допомогою функції **filter**. Однак швидкість обчислень при фільтрації за допомогою БПФ може виявитися суттєво вищою, особливо якщо довжина вхідного сигналу у багато разів перевищує довжину імпульсної характеристики фільтра (або навпаки).

Функція **filtfilt** дозволяє компенсувати фазовий зсув, який вноситься при звичайній фільтрації (іншими словами, ця функція реалізує фільтрацію без внесення тимчасової затримки). Здійснюється це шляхом двонаправленої обробки сигналу. Перший прохід фільтрації здійснюється звичайним чином,

а потім отриманий вихідний сигнал фільтрується вдруге від кінця до початку. За рахунок цього відбувається компенсація фазових зрушень, а результуючий порядок фільтра збільшується вдвічі. Слід зазначити, що з результуючого (еквівалентного двом проходом фільтрації) фільтра не виконується умова причинності.

При практичній реалізації рекурсивних фільтрів високого порядку вони часто представляються як послідовно включених секцій другого порядку. Це дозволяє послабити похибки обчислень, що виникають через помилки заокруглення та квантування коефіцієнтів фільтра. Засоби аналізу помилок такого роду зосереджені у пакеті Filter Design, а пакеті Signal Processing є функція **sosfilt**, що дозволяє реалізувати дискретну фільтрацію даних за допомогою фільтра, представленого у вигляді секцій другого порядку.

Ще однією можливою структурою дискретного фільтра є гратчаста структура. Для здійснення фільтрації за допомогою фільтра, представленого в такій формі, призначена функція **latcfilt**.

Функція **medfilt1**, що реалізує одновимірну медіанну фільтрацію, відноситься до нелінійних алгоритмів фільтрації. Сутність її роботи полягає в тому, що до вхідного сигналу застосовується ковзне вікно заданої довжини, відліки в межах вікна впорядковуються і як вихідний відлік повертається значення з середини впорядкованого вікна (або півсума двох найближчих до середини елементів, якщо вікно має парну довжину). Медіанна фільтрація застосовується, наприклад, для усунення імпульсних перешкод (кляцань) під час обробки звукових сигналів. Функція **medfilt2**, що реалізує двовимірний варіант медіанної фільтрації, розташована у пакеті Image Processing.

2.4 Функції синтезу дискретних фільтрів

Під синтезом дискретного фільтра розуміється вибір таких наборів коефіцієнтів $\{a_i\}$ і $\{b_i\}$, у яких характеристики фільтра, що виходить, задовольняють заданим вимогам. Суворо кажучи, завдання проектування

входить і вибір відповідної структури фільтра з урахуванням кінцевої точності обчислень. Це особливо актуально під час реалізації фільтрів з використанням спеціалізованих мікросхем або цифрових сигнальних процесорів. Ефекти, пов'язані з кінцевою точністю обчислень, можна аналізувати за допомогою функцій Filter Design; функціями синтезу фільтрів ці ефекти не враховуються.

У пакеті Signal Processing є багато функцій, реалізують різноманітні алгоритми синтезу дискретних фільтрів.

Методи синтезу дискретних фільтрів можна розділити на великі групи: з використанням і без використання аналогового прототипу. При використанні аналогового фільтра-прототипу необхідно будь-яким чином представляти аналогову функцію передачі, задану в s -області, дискретну функцію передачі, задану в z -області. У пакеті Signal Processing реалізовано два методи такого перетворення: метод інваріантної імпульсної характеристики та метод білінійного z -перетворення. Обидва методи дають рекурсивні дискретні фільтри.

З використанням методу інваріантної імпульсної характеристики відбувається дискретизація імпульсної характеристики аналогового прототипу. Частотна характеристика дискретного фільтра, що отримується, відповідно, являє собою періодично повторену частотну характеристику аналогового прототипу. Тому цей метод непридатний для синтезу фільтрів верхніх частот і взагалі фільтрів, коефіцієнт передачі яких не прагне нуля зі зростанням частоти. Метод інваріантної імпульсної характеристики реалізований у пакеті Signal Processing за допомогою функції **impinvar**.

З використанням методу білінійного z -перетворення відбувається спотворення характеристики аналогового прототипу лише вздовж частотної осі. При цьому частотний діапазон аналогового фільтра (від нуля до нескінченності) перетворюється на робочий діапазон частот дискретного фільтра (від нуля до половини частоти дискретизації). Перетворення частотної осі описується функцією арктангенса, тому частоти значно менші

частоти дискретизації перетворюються приблизно лінійно. Цей метод реалізується за допомогою функції `bilinear` для довільного аналогового прототипу. Крім того, є готові функції розрахунку фільтрів нижніх і верхніх частот, смугових та режекторних фільтрів методом білінійного z -перетворення за аналоговими прототипами з АЧХ Баттерворта, Чебишева першого та другого роду, а також Кауера (еліптичні фільтри). Це відповідно функції **`butter`**, **`cheby1`**, **`cheby2`** та **`ellip`**. Всі ці функції можуть використовуватись і для розрахунку аналогових фільтрів (див. раніше). Ознакою дискретного варіанта розрахунку є рядок «`s`» у списку вхідних параметрів. Є також функції визначення необхідного порядку цих фільтрів за заданими параметрами АЧХ (граничних частот смуг пропускання і затримування, і навіть допустимим пульсаціям у цих полосах). Це відповідно функції **`buttord`**, **`cheb1ord`**, **`cheb2ord`**, **`ellipord`**. Так само, як і функції синтезу фільтрів, ці функції дозволяють визначати необхідний порядок і для аналогових фільтрів. Ознакою дискретного варіанта розрахунку є рядок 's' у списку вхідних параметрів.

Як приклад синтезуємо еліптичний ФНЧ четвертого порядку з такими ж параметрами, як у аналогового фільтра в одному з попередніх прикладах (частота зрізу 3 кГц, пульсації АЧХ у смузі пропускання 1 дБ і придушення сигналу в смузі затримування 20 дБ). Частоту дискретизації приймемо 12 кГц. Після синтезу побудуємо графіки АЧХ та ФЧХ отриманого фільтра за допомогою функції **`freqz`**.

```
Fs = 12000;           % частота дискретизації
F0 = 3000;          % частота зрізу
[b, a] = ellip(4, 1, 20, F0/Fs*2); % розрахунок фільтра
freqz(b, a, [], Fs); % виведення графіків
```

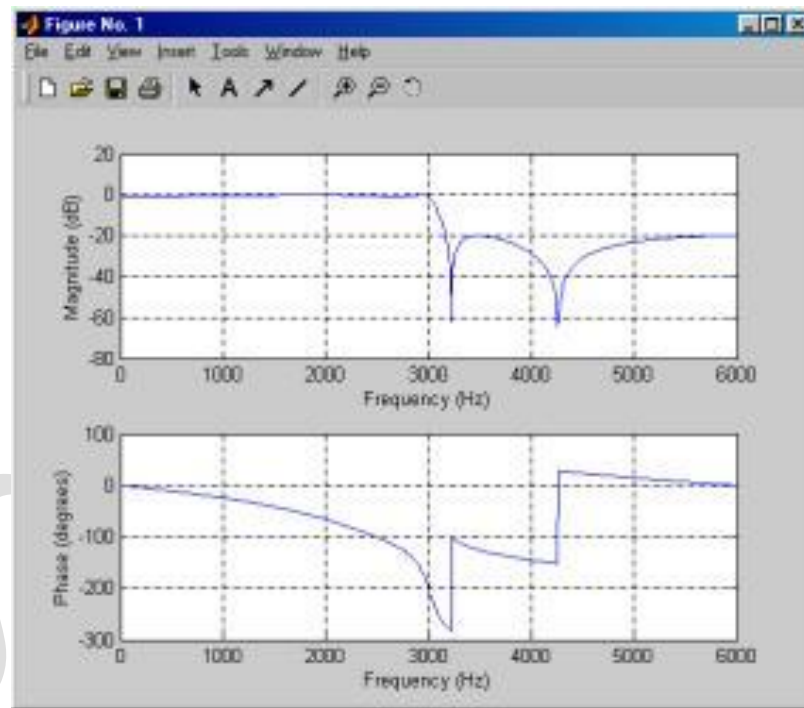


Рисунок 2.3 – АЧХ і ФЧХ фільтра

Методи синтезу, які не використовують аналоговий прототип, називаються прямими. Їх, своєю чергою, можна розділити на дві групи: методи синтезу рекурсивних і нерекурсивних фільтрів.

До функцій прямого синтезу нерекурсивних фільтрів належать такі:

1) Функції, що реалізують синтез фільтрів шляхом зворотного перетворення Фур'є від бажаної АЧХ з подальшим множенням імпульсної характеристики, що вийшла, на деяку вагову функцію (вікно) для послаблення пульсацій АЧХ, що з'являються через ефект Гіббса. Це функції `fir1` та `fir2`. Сюди можна віднести функцію синтезу ФНЧ з косинусоїдальним згладжуванням АЧХ - **`firrcos`**.

2) Функції, що реалізують мінімізацію середньоквадратичного відхилення АЧХ фільтра, що виходить від заданої. Це функції **`firls`**, **`fircls`** та **`fircls1`**. Останні дві функції вирішують оптимізаційне завдання з обмеженням максимального відхилення АЧХ від заданої. Це дозволяє уникнути появи великих викидів АЧХ поблизу перехідних смуг.

3) Функції, що реалізують мінімаксну оптимізацію, тобто мінімізацію пікового відхилення АЧХ фільтра, що виходить від заданої. В результаті утворюються фільтри з рівномірними пульсаціями АЧХ. До цієї групи належать функції `remez` (стандартний варіант методу Ремеза, реалізований ще перших версіях пакета `Signal Processing`) і `sremez` (розширений варіант, що підтримує синтез фільтрів з нелінійної ФЧХ і з комплексними коефіцієнтами). Крім того, функція `remezord` дозволяє за заданими параметрами АЧХ оцінити необхідний порядок фільтра під час синтезу методом Ремеза.

До функцій прямого синтезу рекурсивних фільтрів належать такі:

- **yulewalk** – синтез рекурсивного фільтра з довільною шматково-лінійною АЧХ методом Юла-Уолкера.

- **invfreqz** - дана функція призначена на вирішення завдання ідентифікації систем, вона дозволяє визначити коефіцієнти чисельника і знаменника функції передачі дискретної системи з набору значень цієї функції передачі різних частотах.

Крім пакета `Signal Processing`, ряд функцій синтезу дискретних фільтрів є у пакетах `Communications` та `Filter Design`.

2.5 Функції спектрального аналізу і статистичної обробки сигналів

Слова "спектральний аналіз" у свідомості багатьох користувачів `MATLAB` міцно асоціюються з функцією `fft`, що виконує дискретне перетворення Фур'є (ДПФ). Однак це лише взаємно-однозначне лінійне перетворення, що дає уявлення детермінованого сигналу в частотній області. Якщо ж аналізований сигнал є випадковим, йому має сенс лише оцінка спектральної щільності потужності, до розрахунку якої доводиться тим чи іншим способом виконувати усереднення наявних даних. Крім того, у ряді випадків нам відома деяка додаткова інформація про аналізований сигнал, і цю інформацію бажано врахувати при спектральному аналізі.

Методи спектрального аналізу випадкових сигналів поділяються на два великі класи — непараметричні та параметричні. У непараметричних (nonparametric) методах використовується лише інформація, що міститься у відліках аналізованого сигналу. Параметричні (parametric) методи припускають наявність деякої статистичної моделі випадкового сигналу, а процес спектрального аналізу в даному випадку включає визначення параметрів цієї моделі. Використовується термін "модельний спектральний аналіз" (Model-Based Spectrum Analysis, MBSA).

У пакеті Signal Processing є функції, що реалізують різноманітні методи спектрального аналізу як параметричні, так і непараметричні (необхідно ще раз підкреслити, що під спектральним аналізом тут мається на увазі оцінка спектральної щільності потужності випадкового процесу). Крім того, є функції отримання інших усереднених характеристик випадкових дискретних сигналів.

2.5.1 Спектр дискретного випадкового процесу

Для визначення спектральних характеристик дискретного випадкового процесу обчислюється середній спектр потужності його обмеженого довжиною фрагмента, а потім довжина фрагмента спрямовується до нескінченності:

$$W(\omega) = \lim_{n \rightarrow \infty} \frac{1}{2n+1} \left| \sum_{k=-n}^n x(k) e^{-j\omega k T} \right|^2. \quad (2.3)$$

Тут $x(k)$ – відліки випадкового процесу, T – період дискретизації.

Крім того, цей спектр можна виразити через кореляційну функцію випадкового процесу:

$$W(\omega) = \sum_{k=-\infty}^{\infty} R_x(k) e^{-j\omega k T}. \quad (2.4)$$

Цей вираз є дискретним аналогом теореми Вінера-Хінчина: спектр дискретного випадкового процесу є перетворенням Фур'є від його кореляційної функції.

2.5.2 Непараметричні методи

Як мовилося раніше, з використанням непараметричних методів розрахунку спектру випадкового процесу використовується лише інформація, укладена у відліках сигналу, без будь-яких додаткових припущень. У пакеті Signal Processing реалізовано три такі методи — періодограма, метод Уелча (Welch) та метод Томсона (Thomson).

Періодограмою (periodogram) називається оцінка спектральної щільності потужності, отримана за N відліками однієї реалізації випадкового процесу згідно з визначенням (2.3) (природно, не шляхом взяття межі, а усереднення кінцевого числа доданків). Якщо під час розрахунку спектра використовується вагова функція (вікно), отримана оцінка спектра потужності називається модифікованою періодограмою (modified periodogram):

Співвідношення (2.4) виконується тільки при нескінченному числі відліків, що використовуються, тому при будь-якому кінцевому N періодограма оцінка спектральної щільності потужності виявляється зміщеною - виходить, що всередині суми (2.4) кореляційна функція сигналу множиться на трикутну вагову функцію. Крім того, можна показати, що періодограма не є заможною оцінкою спектральної щільності потужності, оскільки дисперсія такої оцінки порівнянна з квадратом її математичного очікування за будь-якого N . Зі зростанням числа використовуваних відліків значення періодограми починають все швидше флюктувати — її графік стає дедалі більш порізаним.

У пакеті Signal Processing обчислення періодограми (зокрема модифікованої) провадиться за допомогою функції **periodogram**.

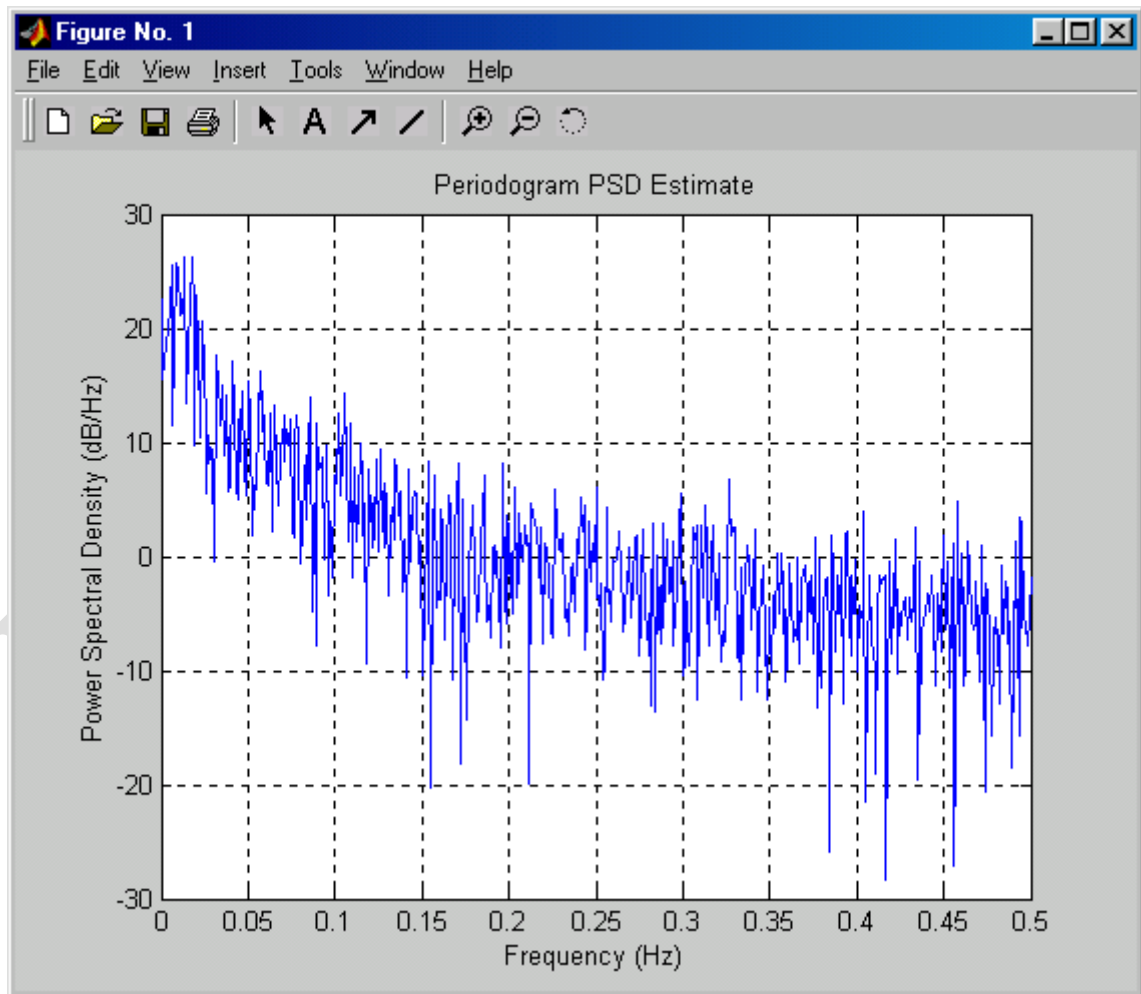


Рисунок 2.4 – Періодограма

Для зменшення порізаності періодограми необхідно застосувати якийсь усереднення. Бартлетт (Bartlett) запропонував розділяти аналізований сигнал на сегменти, що не перекриваються, обчислювати для кожного сегмента періодограму і потім ці періодограми усереднювати. Якщо кореляційна функція сигналу на тривалості сегмента згасає до мало малих значень, то періодограми окремих сегментів можна вважати незалежними. Уелч (Welch) вніс у метод Бартлетта два удосконалення: використання вагової функції і розбиття сигналу на фрагменти, що перекриваються. Застосування вагової функції дозволяє послабити розтікання спектру і зменшити зміщення одержуваної оцінки спектра щільності потужності ціною незначного погіршення роздільної здатності. Перекриття сегментів введено для того, щоб збільшити їх кількість та зменшити дисперсію оцінки.

Обчислення при використанні методу Уелча (він називається ще методом усереднення модифікованих періодограм - averaged modified periodogram method) організуються наступним чином: вектор відліків сигналу ділиться на сегменти, що перекриваються, кожен сегмент множиться на використовувану вагову функцію, для зважених сегментів періодограми, періодограми всіх сегментів усереднюються.

Метод Уелча є найпопулярнішим періодограмним методом спектрального аналізу. У пакеті Signal Processing він реалізується за допомогою функції `pwelch`.

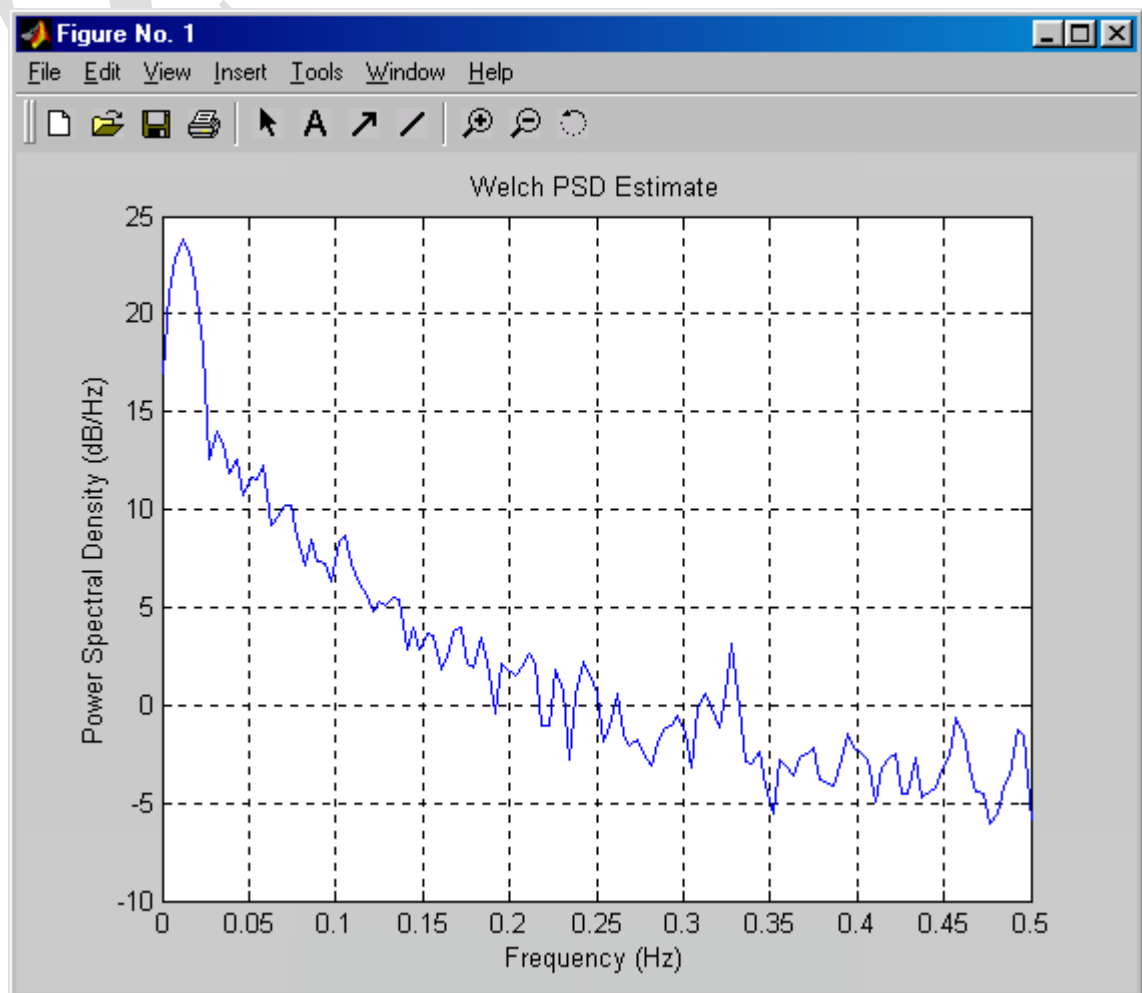


Рисунок 2.5 – Оцінка спектру методом Уелча

Метод Томсона, реалізований функцією `pmtm`, заснований на використанні витягнутих сфероїдальних функцій (prolate spheroidal

functions). Ці функції кінцевої тривалості забезпечують максимальну концентрацію енергії у заданій смузі частот. Крім спектральної оцінки, функція `pmtm` може повертати її довірчий інтервал.

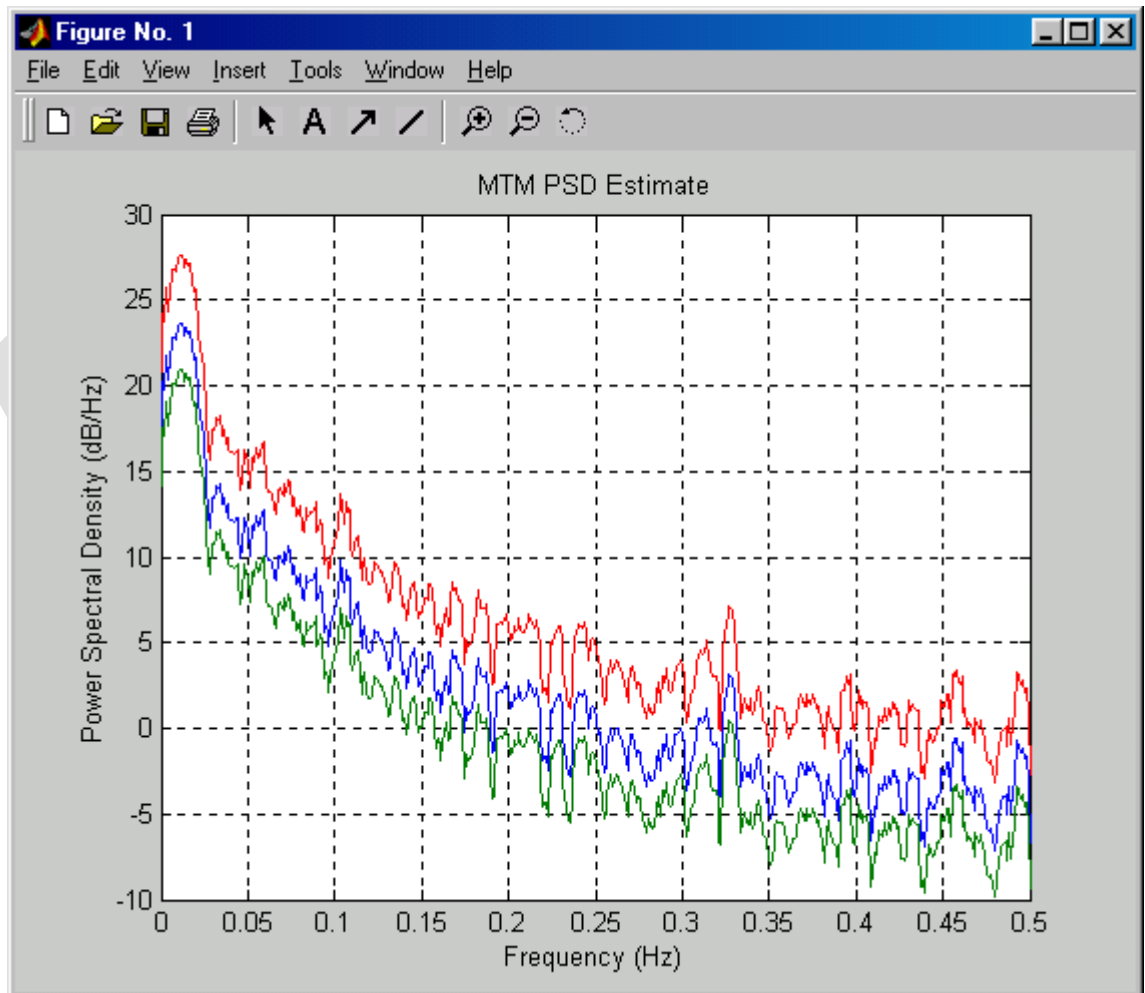


Рисунок 2.6 – Оцінка спектру методом Томпсона

На виведеному функцією `pmtm` графіку разом із оцінкою спектра потужності показані межі довірчого інтервалу.

2.5.3 Параметричні методи

Використання параметричних методів має на увазі наявність деякої математичної моделі аналізованого випадкового процесу. Спектральний аналіз зводиться в даному випадку до вирішення оптимізаційної задачі, тобто

пошуку таких параметрів моделі, при яких вона найбільш близька до сигналу, що реально спостерігається.

У пакеті Signal Processing реалізовано ряд різновидів авторегресійного аналізу та два методи, засновані на аналізі власних чисел та векторів кореляційної матриці сигналу: MUSIC (MUltiple SIgnal Classification) та EV (EigenVectors).

2.5.4 Авторегресійні методи

Відповідно до авторегресійної моделі сигнал формується шляхом пропускання дискретного білого шуму через "чисто рекурсивний" фільтр N -го порядку. Спектральна густина потужності такого сигналу пропорційна квадрату модуля коефіцієнта функції передачі формуючого фільтра. Таким чином, даний метод спектрального аналізу зводиться до визначення коефіцієнтів фільтра заданого порядку, оцінки потужності збудливого білого шуму та аналітичного розрахунку спектральної щільності потужності. Для визначення коефіцієнтів моделі провадиться мінімізація помилки лінійного передбачення сигналу. Теоретичний аналіз показує, що оптимальні коефіцієнти моделі визначаються кореляційною функцією сигналу.

Авторегресійні методи аналізу спектра найбільше підходять для сигналів, що є авторегресивними процесами. Взагалі хороші результати ці методи дають тоді, коли спектр аналізованого сигналу має чітко виражені піки. Зокрема, таких сигналів відноситься сума декількох синусоїд з шумом.

При використанні авторегресійних методів важливо правильно вибрати порядок авторегресійної моделі — він повинен бути вдвічі більшим за кількість синусоїдальних коливань, які, ймовірно, містяться в аналізованому сигналі.

2.6 Функції параметричного моделювання

Під параметричним моделюванням розуміються вибір деякої математичної моделі випадкового процесу і наступний підбір параметрів цієї моделі для забезпечення максимальної відповідності між сигналом, що формується моделлю, і наявною реальною вибіркою даних.

Однією з широко використовуваних практично є авторегрессионная (AR) модель, у якій випадковий сигнал формується шляхом пропускання дискретного білого шуму через “чисто рекурсивний” (тобто використовує затриманих відліків вхідного сигналу) формуючий фільтр. Чотири функції пакету Signal Processing - **arburg**, **arcov**, **armcov** та **aryule** - призначені для отримання оцінок коефіцієнтів формуючого фільтра та дисперсії (потужності) збуджуючого фільтр білого шуму.

Якщо в розпорядженні є оцінка комплексного коефіцієнта передачі системи на різних частотах, можна побудувати модель системи, частотна характеристика якої буде максимально близькою до вимірної. Під реалізованістю системи тут мається на увазі представність її функції передачі у вигляді дробно-раціональної функції із заданими порядками поліномів чисельника та знаменника. Параметричне моделювання у разі зводиться до знаходження оптимальних коефіцієнтів поліномів чисельника і знаменника функції передачі. Це завдання вирішується двома функціями пакета Signal Processing: функція **invfreqs** дозволяє побудувати модель аналогової системи, а функція **invfreqz** виконує аналогічну операцію стосовно дискретних систем.

Ще один варіант задачі параметричного моделювання передбачає побудову моделі системи за оцінкою її імпульсної характеристики. Для цього у пакеті Signal Processing є дві функції. Функція **prony** використовує те що, що імпульсна характеристика рекурсивної дискретної системи за відсутності в неї кратних полюсів є суму дискретних експоненційних функцій (загалом комплексних). Стійкість отриманої системи не гарантується, проте перші n

відліків (n - заданий при розрахунку порядок чисельника функції передачі системи) її імпульсної характеристики точно збігаються із заданими.

Друга функція моделювання системи за імпульсною характеристикою – функція **stmcb** – не прагне забезпечити точне збіг початкових фрагментів імпульсних характеристик – натомість вона мінімізує квадратичне відхилення отриманої характеристики від заданої, тобто суму квадратів модулів різниць відліків отриманої та бажаної імпульс. Функція реалізує ітераційний метод Штейгліца-МакБрайда, який зводиться до багаторазового вирішення системи лінійних рівнянь щодо коефіцієнтів поліномів функції передачі системи, що шукається.

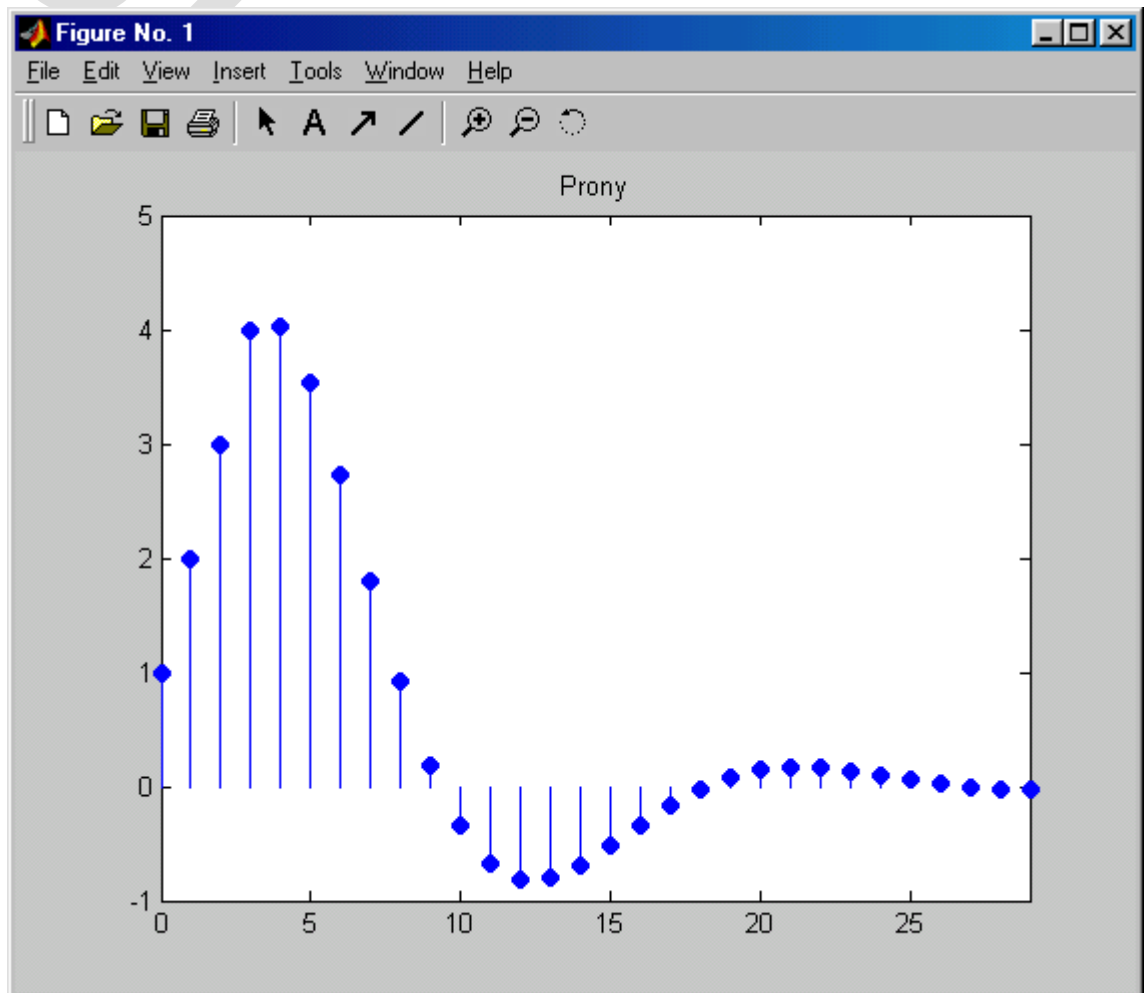


Рисунок 2.7 – Модель системи, отримана методом Проні

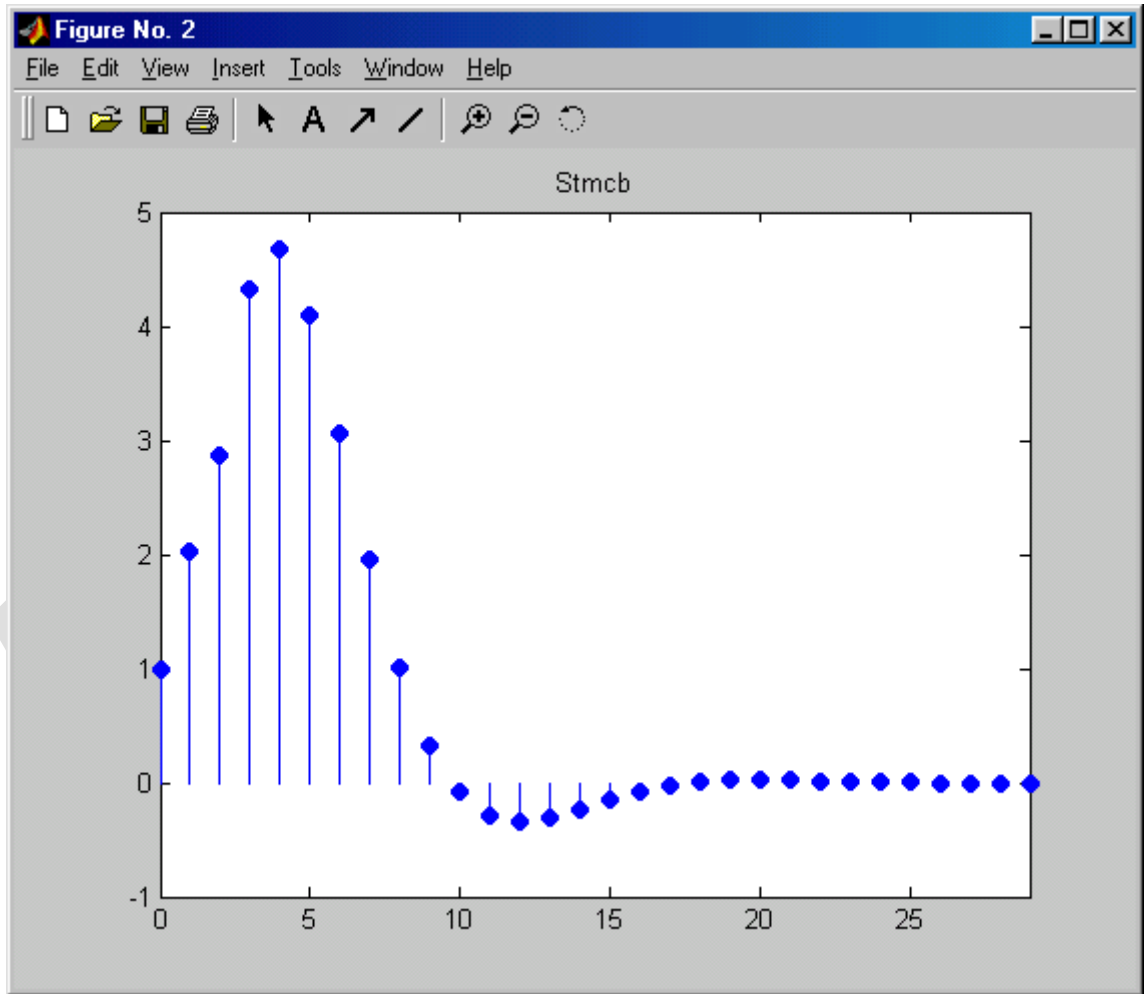


Рисунок 2.8 – Модель системи, отримана методом Штейгліца - МакБрайда

Порівняння графіків наочно демонструє різницю двох алгоритмів. При використанні методу Проні перші чотири відліки отриманої імпульсної характеристики точно збігаються із заданими, проте надалі відхилення від заданих величин сильно зростають, а після закінчення заданого фрагмента спостерігається "хвіст" з досить великим рівнем, тому що функція `prony` не робить жодних припущень про необхідні значення імпульсної характеристики поза заданого фрагмента. Функція **stmcb** мінімізує квадратичну помилку відтворення заданої нескінченної імпульсної характеристики, при цьому після закінчення явно заданого фрагмента вона вважається рівною нулю. В результаті точної відповідності відліків заданої та отриманої імпульсних характеристик не спостерігається (за винятком

першого), проте помилка відтворення характеристики "розмазана" за відліками більш рівномірно.

2.7 Допоміжні функції

Ця група містить досить багато функцій. Чимало їх призначені насамперед "для внутрішнього використання" — вони викликаються іншими функціями пакета. Проте низка функцій цієї категорії має цілком самостійну цінність.

Функція **buffer** дозволяє представляти вектор відліків сигналу матрицю послідовних кадрів, причому ці кадри можуть перекриватися.

Функції **mod** та **demod** здійснюють відповідно модуляцію та демодуляцію. Підтримуються такі види модуляції: амплітудна, амплітудна з пригніченою несучою, односмугова, фазова, частотна, квадратурна, широтно-імпульсна, час-імпульсна.

Функції **uencode** та **uddecode** реалізують відповідно рівномірне квантування та відновлення сигналу за номерами рівнів квантування.

Функція **strips** призначена для виведення графіка сигналу кілька рядків. Це буває корисно, якщо необхідно охопити поглядом довгий сигнал цілком, а роздільна здатність вертикальної осі не має великого значення.

Сімейство функцій, імена яких починаються із символів **dpss**, призначено для розрахунку дискретних витягнутих сфероїдальних функцій та роботи з базою даних, призначеної для зберігання розрахованих функцій. Дискретні витягнуті сфероїдальні функції використовуються при спектральному аналізі методом Томсона (див. вище розділ "Функції спектрального аналізу та статистичної обробки сигналів", підрозділ "Непараметричні методи").

Функції **cell2sos** і **sos2cell** дозволяють зберігати інформацію про фільтри, представленому у вигляді послідовно включених секцій другого порядку, не тільки у вигляді матриці, але й у вигляді масиву осередків. Ці дві

функції здійснюють перетворення уявлення між зазначеними двома формами.

Функція **specgram** здійснює розрахунок спектрограми сигналу, тобто спектрів послідовних фрагментів сигналу, що виділяються за допомогою ковзного вікна. Результати обчислень можуть повертатися як матриці або відображатися кольором в координатах “час-частота”.

Функція **cplxpair** виділяє комплексно-сполучені пари у векторах комплексних чисел.

Функція **eqtlength** дозволяє вирівняти довжини двох векторів шляхом доповнення коротшого з них нулями в кінці.

Функція **seqperiod** призначена для перевірки елементів вектора на періодичність та визначення періоду їх повторення.

3 ОТРИМАННЯ ДАНИХ З ЗОВНІШНІХ ДЖЕРЕЛ

3.1 Читання wav-файлів

Перетворення аналогового сигналу на цифровий і назад – це процеси, що виконуються апаратними засобами. MATLAB ж, будучи програмним продуктом, може лише взаємодіяти з відповідним обладнанням (така взаємодія здійснюється, наприклад, за допомогою Data Acquisition). Крім того, у MATLAB передбачено засоби для відтворення та запису звуку, а також для роботи зі звуковими файлами формату wav. У цьому розділі ми розглянемо зчитування та запис wav-файлів та відтворення звуку.

Для зчитування wav-файлів у MATLAB використовується функція **wavread**. У найпростішому випадку вона може бути використана наступним чином: `y = wavread('filename');`. Тут `filename` - ім'я звукового файлу (розширення `.wav` вказувати не обов'язково). У ім'я файлу необхідно включати повний шлях, за винятком тих випадків, коли файл знаходиться в поточному каталозі або одному з каталогів, що входять до списку пошуку MATLAB.

В результаті виклику функції в змінну `y` поміститься весь вміст зазначеного файлу. Рядки матриці відповідають відлікам сигналу, стовпці - каналам, яких у wav-файлі може бути кілька.

У звукових файлах відліки сигналу представлені цілими числами, що у діапазоні `-128...+127` (8 біт на відлік) чи `-32768...+32767` (16 біт на відлік). MATLAB нормує ці значення, зводячи їх до діапазону `-1...+1`. Крім власне відліків сигналу, у wav-файлах зберігається ще й службова інформація про частоту дискретизації, кількість бітів на відлік і т.п. Дізнатися частоту дискретизації можна, використовуючи під час виклику функції другий вихідний параметр: `[y, Fs] = wavread('filename');`. При цьому змінна `Fs` набуває значення, що дорівнює частоті дискретизації в герцах. Щоб дізнатися, скільки рівнів сигналу міститься у звуковому файлі (точніше,

числа бітів на відлік), необхідно додати третій вихідний параметр: `[y, Fs, bits] = wavread('filename');`.

Ще два параметри, які хочеться заздалегідь знати, - це число відліків і каналів запису. Для отримання інформації необхідно викликати функцію `wavread` з двома вхідними параметрами. Перший – це, як і раніше, ім'я файлу, другим же має бути текстовий рядок `'size': wavesize = wavread('filename', 'size');`.

На відміну від попередніх варіантів виклику функції, в даному випадку не зчитується самих звукових даних. З `wav`-файлу лише вилучається службова інформація, яка повертається як двоелементного вектора-рядка (у наведеному прикладі - `wavesize`). Перший елемент вектора містить кількість відліків, другий – кількість каналів. Нарешті, є й найважливіша можливість зчитування даних із `wav`-файлу не повністю, а окремими фрагментами (без цього не можна було б працювати з великими файлами). Для цього також використовується другий вхідний параметр функції `wavread`. Якщо цей параметр є числом, буде зчитано відповідну кількість відліків (починаючи з першого): `y = wavread('filename', N);` Якщо потрібний фрагмент розташований не на початку файлу, доведеться вказати його початок і кінець. У цьому випадку другий вхідний параметр функції `wavread` повинен бути двоелементним вектором: `y = wavread('filename', [n1 n2]);` У результаті змінну `y` будуть раховані відліки з номерами від `n1` до `n2` включно (нумерація відліків, як і елементів матриць у `MATLAB`, починається з одиниці). При цьому прочитуються всі канали звукозапису. Можливості зчитування інформації з окремих каналів не передбачено.

Розглянемо роботу з `wav`-файлами на конкретному прикладі. Візьмемо для цього бадьорий акорд `tada.wav`, який знаходиться в стандартному наборі звуків `Windows`. Насамперед дізнаємося кількість відліків та каналів:

```
>>wavesize = wavread('c:\windows\media\tada', 'size')
wavesize =
42752      2
```

Як бачимо, цей стереозапис (2 канали) містить 42752 відліки.

Далі отримуємо інформацію про частоту дискретизації та кількість біт на відлік:

```
>>[y, Fs, bits] = wavread('c:\windows\media\tada', 1)
```

```
y =
```

```
0 0
```

```
Fs =
```

```
22050
```

```
bits =
```

```
16
```

Отримані результати очевидні – частота дискретизації 22,05 кГц, використовується 16 біт /відлік (65536 рівнів сигналу). "У навантаження" ми отримали значення першого відліку з обох каналів (вектор y). Тепер ми можемо дізнатися про файл ще дещо, наприклад, визначити тривалість звучання. Для цього потрібно розділити число відліків (перший елемент раніше отриманого вектора `wavsize`) на частоту дискретизації F_s :

```
>> wavsize(1)/Fs
```

```
ans =
```

```
1.9389
```

Результат – тривалість звуку за секунди. Щоб дізнатися, скільки пам'яті потрібно MATLAB для зберігання всього запису, потрібно перемножити число відліків і кількість каналів, а потім збільшити результат ще у 8 разів. Можна відразу ж поділити отримане значення на 1024 щоб отримати відповідь в кілобайтах, або на 10242, якщо очікуване число краще вимірювати мегабайтами:

```
>>prod(wavsize)*8/1024
```

```
ans =
```

```
668
```

Витрата пам'яті виявляється цілком помірною (668 Кбайт), так що можемо дозволити собі зберігати файл у пам'яті повністю:

```
>> y = wavread('c:\windows\media\tada');
```

Тепер весь вміст файла зчитано в матрицю *y*. Перевіримо її розміри:

```
>>size(y)
```

```
ans =
```

```
42752      2
```

Як видно з відповіді, розміри збігаються з отриманою раніше інформацією про файл: 42752 відліків (рядків), два канали (стовпця). Щоб переглянути звуковий сигнал, виведемо його у вигляді графіка – окремо для правого та лівого каналів стереозапису, використовуючи для цього функцію `subplot` (рисунок 1):

```
>>subplot(2, 1, 1)
```

```
>>plot(y(:, 1))
```

```
>>subplot(2, 1, 2)
```

```
>>plot(y(:, 2))
```

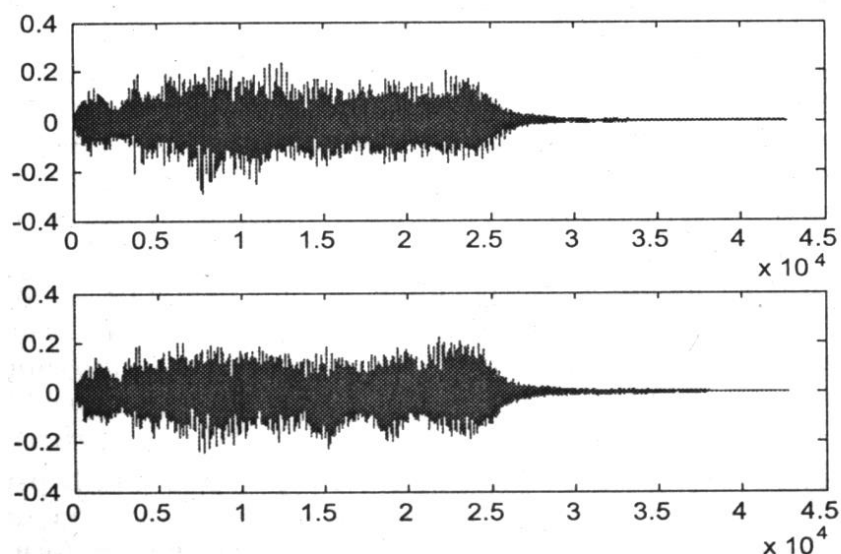


Рисунок 3.1 – Графік звукового сигналу

Якщо при виведенні графіка роздільна здатність по горизонталі має більше значення, ніж по вертикалі, можна скористатися функцією `strips`, спеціально призначеною для відображення довгих сигналів у нарізаному вигляді (фрагменти виводяться один під одним). Синтаксис виклику `strips` є наступним: `strips(x, N)`. Тут x – вектор відліків сигналу (двовимірні масиви не допускаються), N – число відліків у кожному фрагменті (цей параметр можна опустити, за замовчуванням розмір фрагмента становить 200 відліків). Як приклад виведемо за допомогою `strips` графік (лівого) каналу сигналу `tada.wav` (рисунок 3.2):

```
>> strips(y(:, 1), 10000)
```

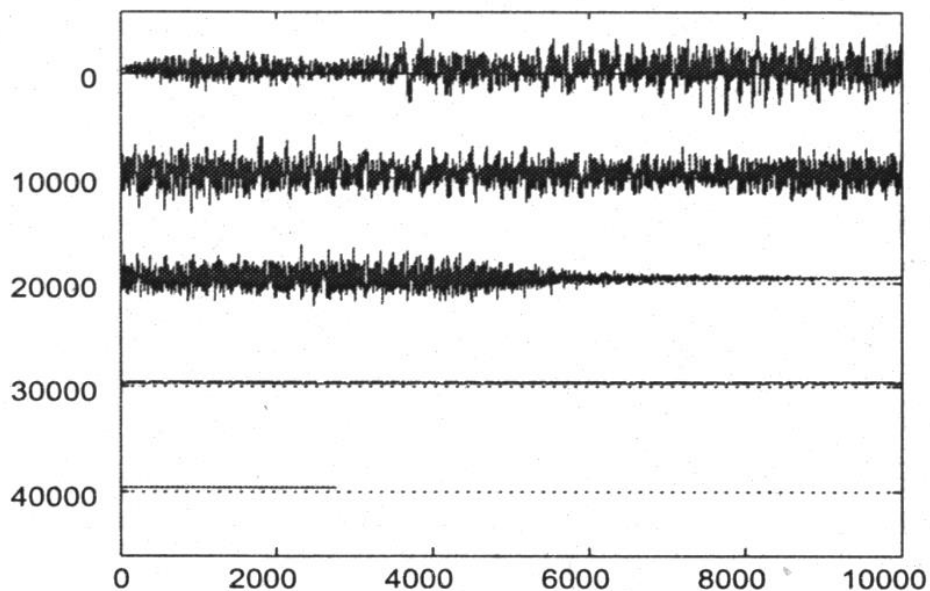


Рисунок 3.2 – Графік сигналу, виведений за допомогою функції **strips**

Над поміщеним у змінну MATLAB звуковим сигналом можна виконувати будь-які перетворення, а потім відтворити отриманий звук або зберегти його у вигляді нового wav-файлу. Також можна зберегти сигнал на диску як змінну MATLAB (у вигляді MAT-файлу). Функція `wavread` не є вбудованою – вона повністю написана мовою MATLAB із використанням засобів роботи з двійковими файлами. MATLAB вміє працювати лише з нестисненими wav-файлами (формат PCM). При спробі зчитати файл, у

якому використано якийсь із методів стиснення інформації, буде видано повідомлення «Data compression format (...) is not supported».

3.2 Запис wav-файлів.

Щоб записати вектор (або матрицю) на диск у вигляді wav-файлу, використовується функція `wavwrite`: `wavwrite(y, Fs, N, 'filename')`. Тут `y` - дані, що записуються (вектор для монофонічного запису, двостовпцева матриця - для створення стереофайлу), `Fs` - частота дискретизації в герцах, `N` - число біт на відлік (8 або 16), `'filename'` - ім'я створюваного файлу. Вихідних параметрів даної функції немає. Параметри `N` і `Fs` можна опускати, при цьому використовуються значення за замовчуванням – `N = 16`, `Fs = 8000`.

```
wavwrite(y, Fs, 'filename')
```

```
wavwrite(y, 'filename')
```

Дані, що записуються, повинні бути речовими і лежати в діапазоні $-1 \dots +1$. Значення, що виходять із цього діапазону, будуть «обрізані» до діапазону $-1 \dots +1$.

3.3 Відтворення звуку

Якщо комп'ютер обладнаний звуковою картою, то, крім роботи з wav-файлами, є можливість відтворення векторів і матриць у звуковому вигляді. Для цього є три функції - **sound**, **soundsc** і **wavplay**.

У найпростішому випадку всі три функції викликаються однаково і забезпечують відтворення вектора (або двостовпцевої матриці для стереозвуку), що містить відліки сигналу, із заданою частотою дискретизації `Fs` (у герцах): `sound(y, Fs)`, `soundsc(y, Fs)`, `wavplay(y, Fs)`.

Відмінність між цими функціями полягає в наданих додаткових можливостях.

3.3.1. Функція звуку.

Функція **sound** забезпечує відтворення сигналу із заданими частотою дискретизації та числом рівнів (біт на відлік): `sound(y, Fs, bits)`. Тут y - вектор або двостовпцева матриця відліків сигналу, F_s - частота дискретизації в герцах, $bits$ - число біт на відлік (8 або 16). Параметри $bits$ і F_s можна опускати, при цьому будуть використовуватися їх значення за замовчуванням: $F_s = 8192$ і $bits == 16$. Відтворювані дані мають бути речовими і лежати в діапазоні $-1...1$. Значення, що виходять із цього діапазону, «обрізаються» і стають рівними ± 1 .

Вихідних параметрів функції немає. Після виклику вона передає вектор y звуковій карті для відтворення і відразу ж, не чекаючи закінчення звуку, повертає керування командним рядком MATLAB.

Якщо наступна команда **sound** буде використана до закінчення попереднього звуку, буде видано повідомлення про помилку `Unable to open sound device` (Неможливо відкрити звуковий пристрій).

3.3.2 Функція `soundsc`

Функція **soundsc** (**sound scaled**) відрізняється від функції **sound** лише тим, що здійснює попереднє масштабування відліків сигналу. Для керування масштабуванням додається четвертий вхідний параметр s_lim : `soundsc(y, Fs, bits, s_lim)`. Тут вхідні параметри y , F_s , $bits$ мають те саме призначення, що й функції **sound**. Параметр s_lim має бути двоелементним вектором $[s_low\ s_high]$, він визначає діапазон значень, який буде лінійно перетворений до інтервалу $-1...1$. Перетворення, таким чином, провадиться за формулою

$$y = \frac{2y - (s_{low} + s_{high})}{s_{high} - s_{low}}$$

При $s_lim = [-1 \ 1]$ функція `soundsc` еквівалентна функції `sound`.

Параметри `s_lim`, `Fs`, `bits` при виклику можна опускати, при цьому використовуються значення за замовчуванням: $Fs = 8192$, $bits = 16$ і $s_lim = [\min(y) \ \max(y)]$. Значення за замовчуванням для `s_lim` забезпечує точне зведення повного діапазону значень сигналу до інтервалу $-1 \dots 1$.

3.3.3 Функція `wavplay`

Нарешті, третя функція, призначена відтворення звуку, має ім'я **wavplay**: `wavplay(y, Fs, 'mode')`. Вхідні параметри `y` і `Fs` мають той же сенс, що й у попередніх функцій, а параметр `mode` управляє режимом відтворення. Цей параметр може приймати два значення:

- `'sync'` - синхронний режим, що означає, що функція поверне керування інтерпретатором MATLAB лише після закінчення звуку;
- `'async'` - асинхронний режим, у якому функція передає дані відтворення звуковим драйверам Windows і відразу ж повертає управління системі MATLAB, не чекаючи закінчення звуку.

Параметри `'mode'` і `Fs` під час виклику можна опускати, у своїй використовуються значення за промовчанням: $Fs = 11025$ і `'mode' = 'async'`. Синхронний режим дозволяє організувати в MATLAB-програмі видачу кількох звуків поспіль, не переймаючись розрахунком і дотриманням тимчасових інтервалів між відповідними командами. В асинхронному режимі можна одночасно з відтворенням звуку продовжувати виконання програми. На жаль, при цьому немає можливості програмним шляхом перевірити, чи відтворення звуку закінчилося.

3.3.4 Запис звуку

Функція **wavrecord** дозволяє записати звук змінну MATLAB з допомогою звукової карти комп'ютера: `y = wavrecord(n, Fs, ch, 'dtype')`. Тут `n`

- число відліків, що записуються, F_s - частота дискретизації в герцах, ch - число каналів запису, 'dtype' - тип записуваних даних.

Повертається результат у - матриця, кожен стовпець якої відповідає одному каналу запису. При стереозаписі перший стовпець – лівий канал, другий – правий. Для параметра dtype можливі наступні значення:

- 'double' 16-бітовий запис, дані масштабуються до діапазону $-1...1$ і подаються в 8-байтовому форматі з плаваючою комою;

- 16-бітовий запис, дані масштабуються до діапазону $-1...1$ і подаються в 4-байтовому форматі з плаваючою комою;

- 16-бітовий запис, дані подаються у двобайтовому цілісному форматі (діапазон $-32\,768...32\,767$);

- 8-бітовий запис, дані подаються в однобайтовому беззнаковому цілісному форматі (діапазон $0...255$, нульовій напрузі на вході відповідає значення 128).

Вхідні параметри 'dtype' ch та F_s можна опускати, при цьому будуть використовуватись їх значення за замовчуванням: $F_s = 11025$, $ch = 1$, 'dtype' = 'double'.

3.3.5 Готові записи сигналів.

У розділі `Aisc1ю` базової бібліотеки `МАТЪАВ` і пакеті розширення `Signal Processing` є кілька готових записів сигналів, збережених як `МАТ`-файли. Ці сигнали використовуються як вихідні дані в прикладах, що наводяться в документації, і в демонстраційних програмах. Вони можуть знадобитися для тестування власних програм та алгоритмів. Відомості про наявні сигнали наведено у таблиці 3.1.

Таблиця 3.1 – Записи стандартних сигналів, що є у MATLAB

Ім'я файла	Пакет	Ім'я змінної	Розмір	Частота дискретизації, Гц	Характер звуку
chirp.mat	MATLAB	y	13129 (1,6 с)	8192	Імпульс зі змінною частотою
gong.mat	MATLAB	y	42028 (5,1 с)	8192	Удар гонга
handel.mat	MATLAB	y	73113 (8,9 с)	8192	Хор
laughter.mat	MATLAB	y	52634 (6,4 с)	8192	Сміх
splat.mat	MATLAB	y	10001 (1,2 с)	8192	Тон, зі змінною частотою, потім звук удару
train.mat	MATLAB	y	12800 (1,5 с)	8192	Гудок поїзда
mtlb.mat	SP	mtlb	4001 (0,54 с)	7418	Вимовлене слово "MATLAB"
vcosig.mat	SP	vcosig	19661 (2,4 с)	8192	Зі своєрідною спектрограмою

4 ПІДВИЩЕННЯ ШВИДКОСТІ І ТОЧНОСТІ РОЗРАХУНКІВ

На точність і швидкість розрахунку моделі Simulink можна впливати багатьма способами, включаючи структуру моделі та її параметри. Вирішальні модулі Simulink працюють точно та ефективно та з параметрами заданими для них “за умовчанням”. Однак для деяких моделей можна досягти кращих результатів за швидкістю та точності, якщо задати більш точно параметри вирішувача диференціальних рівнянь. Також, якщо передбачувана поведінка моделі відома, то можна використовуючи цю інформацію підвищити швидкість та точність розрахунків.

4.1 Підвищення швидкості розрахунку

Мінімальна швидкість моделювання може мати багато причин. Серед них можна виділити основні:

- Модель містить блок MATLAB Fcn. При використанні блоку MATLAB Fcn у моделі Simulink на кожному розрахунковому кроці звертається до інтерпретатора мови MATLAB для виконання розрахунків у цьому блоці. Замість блоку MATLAB Fcn, якщо це можливо, слід використовувати блоки Fcn або Math Function.

- Модель включає S-функцію, написану мовою MATLAB. У цьому випадку відбувається звернення до інтерпретатора мови MATLAB на кожному розрахунковому кроці. Замість MATLAB S-функції більш доцільним було б використання S-функцій написаних мовами C або Fortran і відкомпільованих у машинний код, що виконується у вигляді динамічної бібліотеки.

- Модель містить блок пам'яті Memory. Використання блоку пам'яті змушує вирішальні модулі зі змінним порядком (ode15s і ode113) виконувати зниження порядку до першого кожному розрахунковому кроці.

- Максимальний розмір кроку (Max step size) дуже малий. Якщо цей параметр було змінено, слід спробувати виконати моделювання знову, встановивши цей параметр рівним auto.

- Задано надто високу точність розрахунків. Зазвичай значення абсолютної похибки (Relative tolerance) задане рівним 0.1% достатньо більшості розрахунків. При занадто малих значеннях цього параметра крок розрахунку може бути досить малим, що призведе до уповільнення розрахунків.

- Задано дуже великий інтервал розрахунку за часом. Як правило, при моделюванні динамічних систем перехідні процеси становлять більший інтерес, ніж встановлений режим. Після досягнення встановленого режиму розрахунок можна припинити, оскільки ніяких змін у стані моделі не буде. Бажано заздалегідь оцінити передбачуваний час розрахунку виходячи зі знань про об'єкт, що моделюється.

- Модель може виявитися жорсткою, а вирішувач не призначений для моделювання жорстких систем. Слід спробувати використати методи ode15s або ode23tb та порівняти час розрахунку при вирішенні цими методами.

- У моделі використовуються блоки, крок дискретизації яких (Sample time) не є кратним. У цьому випадку Simulink зменшує крок розрахунку до такого значення, щоб він був кратний кроку дискретизації кожного блоку. Наприклад, якщо крок дискретизації одного блоку дорівнює 0.5, а іншого – 0.7, Simulink встановить максимальне значення кроку розрахунку дорівнює 0.1.

- Модель містить контур алгебри. Алгебраїчні контури розраховуються в Simulink за допомогою ітераційної процедури на кожному етапі розрахунку, що уповільнює загальний час розрахунку.

- Модель має блок Random Number, який передає власний вихідний сигнал на вхід інтегратора (блок Integrator). Переважно використовувати блок Band-Limited White Noise block із бібліотеки Sources.

- Модель включає велику кількість блоків Score. Блоки вимагають значного обсягу пам'яті для зберігання даних, що може призвести до використання комп'ютером віртуальної (дискової) пам'яті та суттєвого уповільнення розрахунків.

- У блоках Score параметр Limit data points to last заданий значно меншим, ніж фактична кількість розрахункових кроків (або прапорець цього параметра знято). У цьому випадку при перевищенні числом кроків значення параметра Limit data points to last для відображення кожної нової розрахункової точки буде виконуватися процедура виділення пам'яті, що суттєво уповільнює швидкість розрахунку. Рекомендується встановити параметр Limit data points to last більшим, ніж фактичне число розрахункових кроків. Має сенс також задати параметр Decimation (проріджування) більшим 1, щоб скоротити кількість збережених блоком Score даних.

Швидкість розрахунку можна підвищити в кілька разів, використовуючи прискорений (Accelerator) режим розрахунку. Це можна зробити за допомогою меню Tools або панелі інструментів. У прискореному режимі розрахунку попередньо проводиться трансляція моделі у виконавчий код (dll-файл), а потім проводиться сам розрахунок. Деякі додаткові витрати на трансляцію з лишком окупаються прискоренням розрахунку моделі. Однак за зміни структури моделі процедура трансляції буде повторена. На жаль, прискорений режим розрахунку не може бути використаний у моделях, що мають алгебраїчні контури.

Істотний вииграш у часі може дати використання дискретних моделей замість безперервних.

Найбільш істотним з точки зору швидкості обчислень може виявитися правильний вибір рівня деталізації моделі. Наприклад, якщо моделювання системи електропостачання міста, навряд чи варто моделювати кожен споживач електричної енергії: електричний двигун, чайник, зварювальний апарат і т.п. Достатнім буде створення узагальнених моделей електричних

споживачів на рівні заводського цеху, житлового будинку, трамвайного парку тощо.

4.2 Підвищення точності розрахунку

Щоб перевірити, чи достатньо точно виконується моделювання, слід провести порівняльні розрахунки з різними значеннями параметра *Relative tolerance* (відносна похибка). Наприклад, можна провести розрахунок із заданим "за умовчанням" значенням цього параметра - $1e-3$ і з меншим ($1e-4$) значенням. Якщо результати розрахунків відрізняються незначно, можна вважати, що знайдене рішення є правильним. Якщо рішення значно відрізняються у початковій стадії, слід задати у вигляді досить малий початковий крок розрахунку (*Initial step size*).

Якщо рішення виявляється нестійким, це може бути викликано такими причинами:

- Система, що моделюється, сама є нестійкою.
- Використовується метод *ode15s*. Слід обмежити порядок величиною 2 або використовувати метод *ode23s*.

Якщо рішення здається не точним:

- Слід задати у явному вигляді параметр *Absolute tolerance* (абсолютна похибка) та виконати низку розрахунків, зменшуючи величину цього параметра.

- Якщо зменшення абсолютної похибки точність розрахунків не покращується, слід зменшити відносну похибку (що призведе до зменшення кроку розрахунку) або явно задавати досить малу величину максимального кроку розрахунку.

ВИСНОВКИ

Віброакустична діагностика є найбільш прийнятною для оцінки технічного стану роторних машин та механізмів. Вона найбільш повно відображає природу явищ, робочих процесів, що протікають в них і використовує як джерело інформації вібросигнал, який має велику інформаційну ємність, малу інерційність, велику швидкість поширення, широкий частотний і динамічний діапазони. Ці якості вібросигналу дозволяють оперативну, а при використанні ЕОМ і в реальному масштабі часу, оцінювати технічний стан об'єкта діагностування.

MATLAB є ідеальним програмним засобом для створення систем вимірювання, а також систем автоматизації управління на основі технології віртуальних приладів. MATLAB-програма в комплексі з такими апаратними засобами, як багатоканальні вимірювальні аналого-цифрові перетворювачі, дозволяє розробляти системи вимірювання, контролю, діагностики та управління практично будь-якої складності.

Вивчення програмних особливостей системи MATLAB показали, що вона є унікальною у своєму роді прикладною системою програмування і моделювання, яка дозволяє вести розрахунки практично в будь-якій галузі науки і техніки. Дуже ефективно її можна використовувати при математичному моделюванні механічних пристроїв і систем та створювати на її основі найрізноманітніші високоефективні віртуальні прилади для вирішення задач обробки та аналізу діагностичної інформації.

ПЕРЕЛІК ПОСИЛАНЬ

1. Лазарев Ю. Ф. Початки програмування у середовищі MatLAB: Навч. посібник. – К.: "Корнійчук", 1999. – 160 с.
2. Лазарев Ю. Ф. MatLAB 5.x. – К.: Издат. группа BHV, 2000. – 384 с.
3. Лазарев Ю. Ф. Моделирование процессов и систем в MATLAB. Учебный курс. – СПб.: Питер; Киев: Издат. группа BHV, 2005. – 512 с.
4. Лазарев Ю. Ф. Моделювання на ЕОМ. Навч. посібник. – К.: Корнійчук, 2007.-290 с.
5. Мартынов Г. Г., Иванов А. П. MATLAB 5.x, вычисления, визуализация, программирование. - М.: "Кудиц-образ", 2000. - 332 с.
6. Гулятьев А. Визуальное моделирование в среде MATLAB: учебный курс. - СПб. : "Питер", 2000. - 430 с.
7. Краснопрошина А. А., Репникова Н. Б., Ильченко А. А. Современный анализ систем управления с применением MATLAB, Simulink, Control System: Учебное пособие. - К.: "Корнійчук", 1999. - 144 с.
8. Мартынов Г. Г., Иванов А. П. MATLAB 5.x, вычисления, визуализация, программирование. - М.: "Кудиц-образ", 2000. - 332 с.
9. Медведев В. С., Потемкин В. Г. Control System Toolbox. MatLAB 5 для студентов. - М.: ДИАЛОГ-МИФИ, 1999. - 287 с.
10. Потемкин В. Г. Система MatLAB: Справ. пособие. - М.: ДИАЛОГ-МИФИ, 1997. - 350 с.
12. Потемкин В. Г., Рудаков П. И. MatLAB 5 для студентов. - 2-е изд., испр. и дополн. - М.: ДИАЛОГ-МИФИ, 1999. - 448 с.
13. Потемкин В. Г. Система инженерных и научных расчетов MatLAB 5.x: - В 2-х т. Том 1. - М.: ДИАЛОГ-МИФИ, 1999. - 366 с.
14. Потемкин В. Г. Система инженерных и научных расчетов MatLAB 5.x: - В 2-х т. Том 2. - М.: ДИАЛОГ-МИФИ, 1999. - 304 с.
15. Рудаков П. И., Сафонов В. И. Обработка сигналов и изображений. MATLAB 5x. - М.: "ДИАЛОГ-МИФИ", 2000. - 413 с.