

ПЕРВИННА ОБРОБКА СТОХАСТИЧНОЇ ВИМІРЮВАЛЬНОЇ ІНФОРМАЦІЇ ЗАСОБАМИ MATLAB

Викладено зміст етапів первинної обробки стохастичної вимірювальної інформації описано інтерфейс програми та наведені фрагменти текстів програми. Описана методика визначення в інтерактивному режимі аналітичних виразів дробово-раціональних функцій, що апроксимують спектральні щільності стохастичних сигналів

Спектральні алгоритми ідентифікації динамічних систем, оцінювання і фільтрації сигналів [1, 2] використовують динамічні характеристики корисних сигналів і завад – спектральні і взаємні спектральні щільності. Такі характеристики отримують в результаті первинної обробки вимірювальної інформації, яка зазвичай представлена у формі числових масивів, що записані у файли. Первинна обробка передбачає такі операції:

- видалення лінійного тренду з масивів даних;
- визначення статистичних характеристик сигналів;
- визначення кореляційних та взаємних кореляційних функцій сигналів;
- визначення спектральних та взаємних спектральних щільностей сигналів у формі числових масивів;
- представлення отриманих динамічних характеристик в графічній формі;
- апроксимація графічних залежностей аналітичними виразами;

Постановка задачі. Метою даної роботи є розробка комп'ютерної програми, за допомогою якої можна виконувати первинну обробку стохастичної вимірювальної інформації. Особливості програми полягають в реалізації інтерактивного алгоритму апроксимації спектральної та взаємної спектральної щільності аналітичними залежностями. Такі залежності являються початковими даними для синтезу спектральних алгоритмів, отже точність апроксимації експериментальних оцінок спектральної та взаємної спектральної щільності сигналів впливає на точність синтезованих спектральних алгоритмів.

Інтерфейс програми. Програма первинної обробки вимірювальної інформації реалізована засобами математичної системи MATLAB, яка має функції для обробки і візуалізації числових масивів даних, а також засоби для розробки графічного інтерфейсу користувача. Для програмування графічного інтерфейсу використана підсистема інтерактивного проектування графічного інтерфейсу GUIDE, що входить до системи MATLAB [3, 4].

Програма складається з fig-файлу спеціального формату, який включає повний опис графічних об'єктів (компонентів) інтерфейсу, та текстового m-файлу з програмним кодом, що реалізує алгоритми обробки даних, а також містить функції-відповіді компонентів інтерфейсу (callback-функції). Інтерфейс має дві групи елементів управління і індикації. Перша призначена для обчислення характеристик процесів, заданих числовими масивами, друга – для інтерактивного підбору аналітичних виразів спектральної і взаємної спектральної щільностей процесів, що досліджуються.

Програмна реалізація алгоритмів обробки даних. На першому етапі роботи у програму вводять числовий масив даних з файлу текстового формату, потім необхідні параметри для обчислення характеристик сигналу.

Масив вхідних даних та параметри, що обчислені, зберігаються в програмі у полях структури handles, яка створюється при запуску програми. Доступ до полів цієї структури мають всі функції обробки даних.

В програмі були реалізовані операції первинної обробки даних, що наведені вище, в яких використовують функції MATLAB:

detrend – видалення лінійного тренду масиву даних;

xcorr – обчислення кореляційної та взаємної кореляційної функцій;

var – обчислення дисперсії масиву;

pwelch – обчислення спектральної щільності методом Уелча [6];

csd – обчислення взаємної спектральної щільності.

Фрагменти коду callback-функції при обчислення спектральної щільності показані на рис. 1. Ця функція виконується при натисканні на відповідну кнопку інтерфейса:

```
function pbtnPsd1_Callback(hObject, eventdata, handles)
x = detrend(handles.data{1}, 'linear');           % видалення тренду
[psd1, frq] = pwelch(handles.data{1}, handles.nwin, ...
                    handles.noverlap, handles.nfft, handles.Fs);
handles.frq = frq;
axes(handles.magnitude); % активізація вікна графіка спектральної щільності
handles.psdmag = loglog(frq, (abs(psd1)));
% обчислення роздільної здатності по частоті та індикація
set(handles.textDeltaF, 'String', num2str((frq(2)-frq(1)), 4));
set(handles.textDsp, 'String', num2str(var(handles.data{1}), 4));
guidata(gcbo, handles)
```

Рис. 1

Методика апроксимації спектральної щільності. В основу апроксимації спектральної щільності аналітичним виразом покладено її представлення як реакції деякого формувального фільтра на вхідний сигнал білого шуму

$$S_{xx}(s) = S_0(s) \cdot H(s) \cdot H(-s) = S_0 \cdot |H(s)|^2,$$

де S_0 – спектральна щільність білого шуму; $H(s)$ – передавальна функція формувального фільтра у формі дробово-раціональної функції змінної $s = j\omega$

$$|H(s)|^2 = \left| \frac{b(s)}{a(s)} \right|^2 = \frac{b(s)b(-s)}{a(s)a(-s)},$$

де $b(s)$, $a(s)$ – поліноми чисельника і знаменника передавальної функції.

Таким чином, апроксимацію спектральної щільності можна виконати підбором структури і параметрів передавальної функції формувального фільтра.

Аналогічно виконується апроксимація взаємних спектральних щільностей, але з урахуванням результатів апроксимації спектральної щільності $S_{xx}(s)$.

При виконанні апроксимації головною задачею дослідника залишається вибір придатного виду функції апроксимації (кількість елементарних ланок формувального фільтра та вид передавальної функції кожної ланки). Для зручності підбору параметрів функції апроксимації спектральної щільності використовуються лінії-маркери у вікні графіка спектральної щільності: горизонтальний маркер показує заданий коефіцієнт передачі формувального фільтра, вертикальні маркери – частоти спряження його ланок. Дослідник змінює параметри функції апроксимації для кожної з ланок переміщенням маркерів або введенням з клавіатури у відповідні текстові вікна інтерфейсу.

Програма автоматично обчислює нові значення функції апроксимації, перебудовує графік цієї функції. Одночасно змінюється і виводиться у відповідні текстові вікна аналітичний вираз, що описує цей графік. Ступінь наближення графіка функції апроксимації до графіка спектральної щільності дослідник має оцінити візуально.

Аналітичний вираз дробово-раціональної функції, що являється апроксимацією спектральної щільності, подається у вигляді

$$K \cdot \frac{|b_1(s)|^2}{|a_1(s)|^2} \cdot \frac{|b_2(s)|^2}{|a_2(s)|^2} \cdot \frac{|b_3(s)|^2}{|a_3(s)|^2},$$

де $b_i(s)$, $a_i(s)$ – поліноми комплексної змінної $s = j\omega$ 1-го чи 2-го порядку, або одиниця.

Для зміни параметрів функції апроксимації розроблені відповідні функції, звернення до яких здійснюється при переміщенні за допомогою курсору миші ліній-маркерів в полі вікна графіків.

Функція `x_start` виконується при натисканні на ліву клавішу миші, якщо курсор знаходиться на відповідній лінії-маркері, щоб це реалізувати треба для лінії-маркера (графічний об'єкт `line`) встановити звернення до функції `x_start`: `hdl = line('ButtonDownFcn', @x_start)`.

Програмний текст функції `x_start` показано на рис 2.

```
function x_start_Callback(hObject, eventdata, handles)
set(gcf,'WindowButtonMotionFcn', @x_move)      % виклик функції переміщення
                                                % маркера X у вікні
set(gcf,'WindowButtonUpFcn', @stop)           % виклик функції при
                                                % відпусканні кнопки миші
setptr(gcf, 'Irdrag');                         % зміна вигляду курсору
```

Рис. 2.

Властивість `'WindowButtonMotionFcn'` графічного вікна (об'єкта `figure`) забезпечує виклик функції `x_move`. Ця функція при переміщенні курсору зчитує його координату X, відображає нове положення лінії маркера в графічному вікні, передає це значення в якості параметра у функцію апроксимації, обчислює нову лінію апроксимації, виводить нові значення сталої часу та частоти у текстові вікна, відображає змінений аналітичний вираз спектральної щільності у текстовому вікні. Програмний код функції `x_move` наведено на рис. 3

```
function x_move(hObject, eventdata, handles)
currPt = get(gca,'CurrentPoint');              % зчитування координат курсора
cX = currPt(1,1)*ones(2,1);                   % формування нових параметрів лінії-маркера
set(gcf,'XData', cX)                          % відображення маркера в новому положенні
set(handles.edF, 'String', num2str(cX(1,1), 3)) % відображення значення частоти
                                                % у текстовому вікні
T = 1/(2*pi*cX(1,1));                          % обчислення сталої часу ланки
set(handles.edT, 'String', num2str(T, 3))      % і відображення її значення
% оператори обчислення апроксимуючої функції, виводу її графіка
% та внесення обчислених параметрів до полів структури handles
DispSpectr(handles)                           % вивід аналітичного виразу спектральної щільності
guidata(gcf, handles)                         % запам'ятовування структури в пам'яті
```

Рис. 3

При відпусканні клавіші миші функція `x_start` здійснює виклик функції `stop`, яка завершує процес підбору параметрів і встановлює стандартний вигляд курсору у формі стрілки. Програмний текст функції наведено на рис 4.

```
function stop(hObject, eventdata, handles)
set(gcf,'WindowButtonMotionFcn','')
set(gcf,'WindowButtonUpFcn','')
setptr(gcf, 'arrow');                          % відновлення стандартного виду курсору
```

Рис. 4

Для виводу аналітичного виразу функції апроксимації у текстове вікно програми використовується функція `DispSpectr`, яка зчитує обчислені коефіцієнти поліномів, що представляють аналітичний вираз спектральної щільності, з відповідного поля структури `handles` та перетворює їх в текстовий рядок. Програмні коди функцій `DispSpectr` наведено на рис 5.

```

function DispSpectr(handles)
% вивод виразу спектральної щільності у текстове вікно
if get(handles.rbtn1, 'Value'); % здійснення виводу параметрів першої
ланки
    p = handles.W1{2}; % зчитування коефіцієнтів
    str = sprintf('%s |^2', poly2str(p,'s')); % перетворення в текстовий рядок
    set(handles.textW1d, 'String', str) % відображення в текстовому вікні
elseif
    % аналогічні оператори для інших ланок
end

```

Рис. 5

Робота програми була перевірена на прикладі апроксимації спектральної щільності процесу, модельованого фільтрацією випадкового сигналу фільтром з відомою передавальною функцією. За допомогою розробленої програми обчислювались спектральні щільності модельованих сигналів та виконувалась їх апроксимація.

Наприклад, апроксимація спектральної щільності процесу, отриманого на виході формульованого фільтра з передавальною функцією $H(s) = \frac{1}{0,2s+1}$, дала результат

$$S_{xx}(s) = 0,019 \frac{1}{|0,19s+1|^2},$$

а апроксимація спектральної щільності процесу, отриманого на виході формульованого фільтра з передавальною функцією $H(s) = \frac{1}{10,04s^2+0,2s+1}$, дала результат

$$S_{xx}(s) = 0,0208 \cdot \frac{1}{|0,039s^2+0,18s+1|^2}.$$

Таким чином, відмінність коефіцієнтів отриманих аналітичних виразів спектральних щільностей від заданих під час моделювання знаходиться у межах похибки обчислення оцінок спектральних щільностей.

Висновки. Обробка стохастичних даних з використанням розробленої програми показала, що процес апроксимації спектральних щільностей стає більш зручним і прискорюється, але тільки при вдалому виборі виду функції апроксимації. Це підтверджує той факт, що процес апроксимації спектральних щільностей являється творчим і потребує відповідного досвіду дослідника і можливо додаткової інформації про процес, що досліджується.

Список літератури

1. Блохін Л.М., Буриченко М.Ю. Статистична динаміка систем управління: Підручник. – К.: НАУ, 2003. – 204 с.
2. Азарсков В.М., Буриченко М.Ю., Савінов О.М., Суцєнко О.А. Аналітичне конструювання вимірювальних систем. Спектральні алгоритми. – К.: НАУ, 2002. – 88 с.
3. MATLAB. Creating Graphical User Interfaces. Version 6. The MathWorks, Inc, 2003. – 184 p.
4. Кетков Ю.Л., Кетков А.Ю., Шульц М.М. MATLAB 6.x: программирование численных методов. – СПб.: БХВ-Петербург, 2004. – 672 с.
5. Марпл С.Л. Цифровой спектральный анализ и его приложения. – М.: Мир, 1990. – 584 с.
6. Дьяконов В.П., Абраменкова И.В. MATLAB. Обработка сигналов и изображений. Специальный справочник. – СПб.: Питер, 2002. – 608 с.
7. Дьяконов В.П., Круглов В.А. Математические пакеты расширения MATLAB. Специальный справочник. – СПб.: Питер, 2001. – 480 с.