AWE and Y parameters

Name: Mohammed AL Shuaili

Date: 4/2/2025

Objectives:

1. Implement Y parameters
2. Complex frequency hopping
3. Final FYP report

One can obtain the unit step response out of a state space model using recursive convolution based on the pole-residue representation of the transfer function as shown in the below code. This is to avoid explicit convolution which is computationally expensive.

```
clear all
clc
% Input state-space matrices
A = [-2 1 0 0; 1 -2 1 0; 0 1 -2 1; 0 0 1 -1];
B = [1; 0; 0; 0];
C = [1; 0; 0; 0];
t = 0:0.1:2;
% impulse response
%{
eat = @(t) expm(A.*t);
y = @(t)mtimes(mtimes(transpose(C),eat(t)),B);
y_values = arrayfun(@(t) y(t), t);
%}

% step response
sys = ss(A,B,transpose(C),0);
[y_theory, t] = step(sys, t); % get the step input response using step
% Determine the order of the system
q = length(B);
% Compute moments
num_moments = 2 * q;
moments = zeros(1, num_moments);
for i = 1:num_moments
    moments(i) = -transpose(C) * (A^(-i)) * B;
end
figure(1);
ii=1;
hold on
% Generalized approximation for all orders
for approx_order = 1:q
    fprintf('Case %d:\n', approx_order);

    % Construct the moment matrix
    moment_matrix = zeros(approx_order);
    Vector_c = -moments(approx_order+1:2*approx_order)';

    for i = 1:approx_order
        moment_matrix(i, :) = moments(i:i+approx_order-1);
    end
    % find b matrix (deno coef)
    b_matrix = inv(moment_matrix)*Vector_c;

    %find the ploes
    poles = roots([transpose(b_matrix) ,1]);

    % determine residuses
    % form the V matrix
    V = zeros(approx_order);
    for i = 1:approx_order
        for j = 1:approx_order
            V(i, j) = 1/poles(j)^(i-1);
```

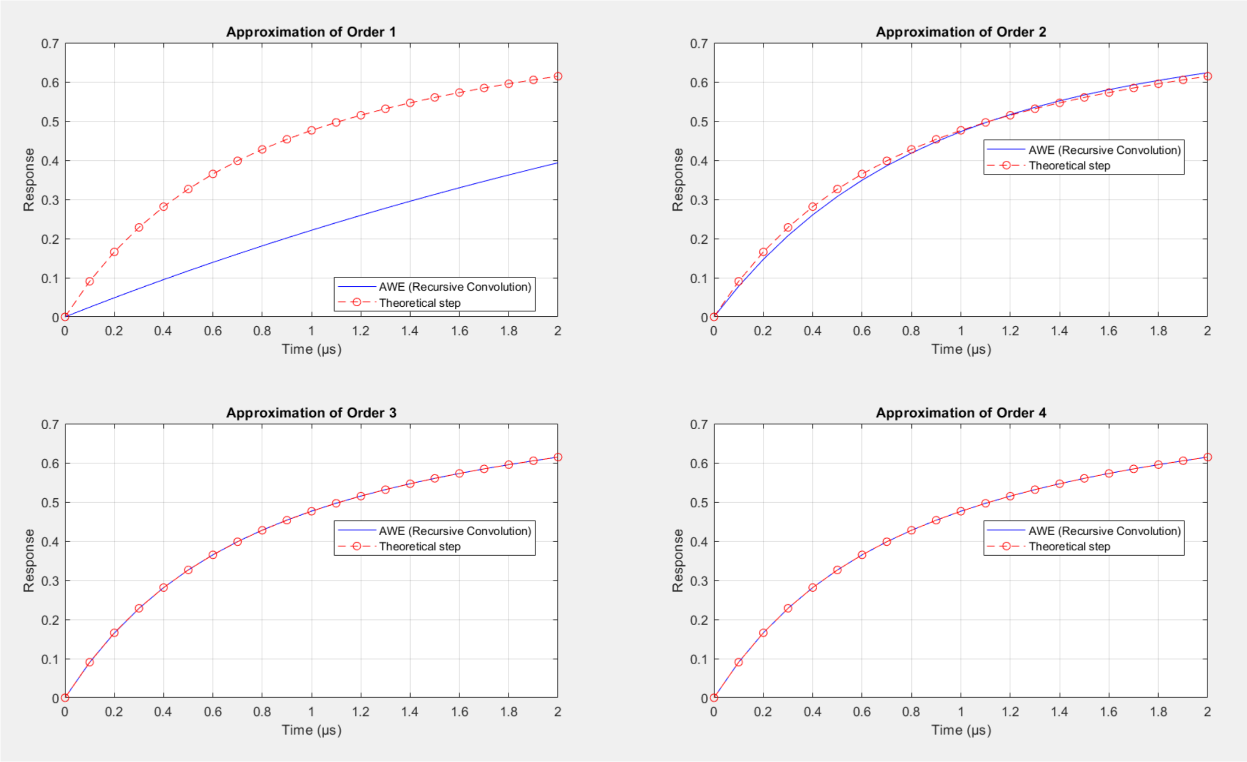
```
        end
    end
    % form the A matrix
    A_diag = diag(1 ./ poles);
    r_moments = moments(1:approx_order); % a helper matrix
    % find the residuse
    residues = -1*inv(A_diag)* inv(V)* transpose(r_moments);
    %form the impulse response
    h =0;
    for i = 1:approx_order
        h = h + residues(i) * exp(poles(i) * t);
    end
    % Recursive Convolution (Section V)

    y_awe = zeros(size(t)); % AWE response

    % y variables for each pole
    y = zeros(length(poles), 1);

    % Recursive convolution loop
    for n = 2:length(t)
    dt = t(n) - t(n-1); % Time step
    exp_term = exp(poles * dt); % Precompute exponentials
        for i = 1:length(poles)
            % Updat state using Eq. (15)
           % y(i) = residues(i) * (1 - exp_term(i)) * 1 + exp_term(i)*y(i);
            y(i) = residues(i) * (1 - exp_term(i))/(-poles(i)) * 1 + exp_term(i)*y(i);
        end
        y_awe(n) = sum(y); % Total response
    end

    % Plot Results
    subplot(2,2,ii)
    ii=ii+1;
    plot(t, y_awe, 'b-'); hold on;
    plot(t, y_theory, 'ro--');
    xlabel('Time (µs)');
    ylabel('Response');
    title(['Approximation of Order ', num2str(approx_order)]);
    legend('AWE (Recursive Convolution)', 'Theoretical step', 'Location', 'Best');
    grid on;
    % plot the output
    %{
    figure(approx_order);
    plot(t,h);
    hold on
    plot(t, y_values,'ro');
    xlabel('Time (\mus)');
    ylabel('V Load (Volts)');
    title(['Apprximation of order',num2str(approx_order)]);
    legend('Awe', 'Theory Impulse', 'Location', 'Best');
    grid on
    %}
end
```



The figure compares step response approximations using the AWE method (blue) with the theoretical response (red circle) for orders 1 to 4. Higher orders improve accuracy, closely matching the theoretical response.

- Y parameters:

$$\begin{bmatrix} I_s \\ -I_R \end{bmatrix} = \begin{bmatrix} Y_{11} & Y_{12} \\ Y_{21} & Y_{22} \end{bmatrix} \begin{bmatrix} V_s \\ V_R \end{bmatrix}$$

Each is approximated with rational function:

$$Y_{ij} = \frac{(a_{nij}s^{n-1} + .. + a_{0ij})}{s^n + .. + b_{0ij}}$$

Considering the open voltage case:

$$\frac{V_R}{V_s} = -\frac{Y_{21}}{Y_{22}} = f\left(\frac{(a_{n21}s^{n-1} + .. + a_{021})}{s^n + .. + b_{021}}, \frac{(a_{n22}s^{n-1} + .. + a_{022})}{s^n + .. + b_{022}}\right) = -\frac{\frac{(a_{n21}s^{n-1} + .. + a_{021})}{s^n + .. + b_{021}}}{\frac{(a_{n22}s^{n-1} + .. + a_{022})}{s^n + .. + b_{022}}}$$

$$= \frac{f_{n-1}s^{n-1} + \ldots + f_0}{s^n + g_{n-1}s^{n-1} + \ldots + g_0}$$

This can then be converted into a state space model as follows:

$$\frac{Z(s)}{V_s} \; and \; \frac{V_R}{Z(s)}$$

$$\boxed{\frac{1}{s^n + g_{n-1}s^{n-1} + \ldots + g_0}} \xrightarrow{Z(s)} \boxed{f_{n-1}s^{n-1} + \ldots + f_0}$$

Then, converting this to time domain gives:

$$\frac{d^n z}{dt^n} + g_{n-1}\frac{d^{n-1}z}{dt^{n-1}} \ldots \ldots + g_0 Z = V_s$$

And,

$$f_{n-1}\frac{d^{n-1}z}{dt^{n-1}} + \cdots .. + f_0 Z = V_R$$

So, let

$$x_1 = Z \, , x_2 = \frac{dz}{dt} , x_3 = \frac{d^2 z}{dt^2} , \ldots .. , x_{n+1} = \frac{d^n z}{dt^n}$$

This gives the A matrix as:

$$A = \begin{bmatrix} 0 & 1 & \ldots . & \ldots & 0 \\ 0 & 0 & 1 & .. & 0 \\ .. & .. & .. & .. & 1 \\ -g_0 & -g_1 & -g_2 & \ldots .. & -g_{n-1} \end{bmatrix}$$

The B matrix:

$$B = \begin{bmatrix} 0 \\ 0 \\ \ldots \\ 1 \end{bmatrix} , (nx1) entries$$

The C matrix:

$$C = [f_0 \quad f_1 \quad \ldots .. \quad f_{n-1}]$$

Let's consider n = 2;

$$\frac{V_R}{V_s} = \frac{f_1 s + f_0}{s^2 + g_1 s + g_0}.$$

The state space model:

$$A = \begin{bmatrix} 0 & 1 \\ -g_0 & -g_1 \end{bmatrix}, \quad B = \begin{bmatrix} 0 \\ 1 \end{bmatrix}, \quad C = [f_0 \quad f_1], \quad D = 0$$

Where,

$$f_{n-1} = -a_{n21}, \quad f_{n-2} = -a_{(n-1)_{21}} - a_{n21}\, b_{(n-1)_{22}}, \quad \ldots. \quad f_0 = -a_{021} b_{022}$$

$$g_{n-1} = b_{(n-1)_{21}} + a_{n22}, \quad g_{n-2} = b_{(n-2)21} + b_{(n-2)21}\, a_{(n-1)22} \ldots., g_0 = b_{021} a_{022}$$

Then AWE can be implemented to get the impulse response and unit step response.

Now, Let's consider the following example to code this in MATLAB, assume the rational approximation for Y_{21} *and* Y_{22}:

$$Y_{21}(s) = \frac{2s+3}{s^2+4s+5}, \qquad Y_{22}(s) = \frac{s+6}{s^2+7s+8}.$$

So,

$$\frac{V_R}{V_s} = -\frac{Y_{21}(s)}{Y_{22}(s)} = -\frac{\frac{2s+3}{s^2+4s+5}}{\frac{s+6}{s^2+7s+8}} = -\frac{(2s+3)(s^2+7s+8)}{(s+6)(s^2+4s+5)}$$

$$\frac{V_R}{V_s} = \frac{-(2s^3+17s^2+37s+24)}{s^3+10s^2+29s+30}$$

To make it strictly proper (numerator degree < denominator degree) we can say:

$$\frac{V_R}{V_s} = -2 + \frac{3s^2+21s+36}{s^3+10s^2+29s+30}.$$

For a third-order system (n=3):

$$H(s) = \frac{f_2 s^2 + f_1 s + f_0}{s^3 + g_2 s^2 + g_1 s + g_0}$$

Do we ignore the (-2s^3) or we only consider the right side of the proper one?

Ignoring the term gives:

$$A = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ -30 & -29 & -10 \end{bmatrix}, \quad B = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}, \quad C = [-24 - 37 - 17]$$

With considering the proper term we get:

$$A = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ -30 & -29 & -10 \end{bmatrix}, \quad B = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}, \quad C = [36 \quad 21 \quad 3], \quad D = -2$$

Code:

depending on the above as obtaining the proper one will depend on the values od Y_{21} and Y_{22}