Include frequency in AWE

Name: Mohammed AL Shuaili

Date: 13/2/2025

Firstly, let's move AWE to a separate file as a function as follows:

```
function [h_impulse, y_step, t] = AWE(A,B,C,D,w,time)
    t = linspace(0,time,250);
    q = length(B);
    num_moments = 2 * q;
    % s=1i*w;
    moments = zeros(1, num_moments);
    for i = 1:num_moments
        moments(i) = -C' * (A)^(-i) * B;
    end
    moments(1)=moments(1)+D;
    approx_order = length(B);

    % 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

    % Solve for denominator coefficients
    b_matrix = inv(moment_matrix) * Vector_c;

    % Compute poles
    poles = roots([b_matrix', 1]);

    % Compute residues
    V = zeros(approx_order);
    for i = 1:approx_order
        for j = 1:approx_order
            V(i, j) = 1 / poles(j)^(i-1);
        end
    end

    A_diag = diag(1 ./ poles);
    r_moments = moments(1:approx_order);
    residues = -1 * (A_diag \ (V \ r_moments'));

    % Impulse response
    h_impulse = zeros(size(t));
    for i = 1:approx_order
        h_impulse = h_impulse + residues(i) * exp(poles(i) * t);
    end

    % Step response using recursive convolution
    y_step = zeros(size(t));
    y = zeros(length(poles), 1);

    for n = 2:length(t)
        dt = t(n) - t(n-1);
        exp_term = exp(poles * dt);
        for i = 1:length(poles)
            y(i) = residues(i) * (1 - exp_term(i))/(-poles(i)) * 1 + exp_term(i) * y(i);
        end
        y_step(n) = sum(y);
    end
end
```

This function will take A,B,C and D matrices as the input and the end point in time, and w(frequency) and will return the impulse and unit responses with respect to t.

- Adjusting AWE to include frequency:

Looking at the general form of Y(s):

$$sX(s) = AX(s) + BU(s)$$

$$Y(s) = C^T X(s)$$

For impulse input, U(s) = 1.

$$Y(s) = C^T (sI - A)^{-1} B$$

Expand this about $s = s_0$.

$$Y(s) = C^T (s_0 I - A)^{-1} B - C^T (s_0 I - A)^{-2} B(s - s_0). + \cdots \dots \dots$$

So,

$$m_0 = C(s_0 I - A)^{-1} B$$

$$m_1 = Y'(s_0) = -C^T (s_0 I - A)^{-2} B$$

$$m_2 = \frac{Y''(s_0)}{2!} = C^T (s_0 I - A)^{-3} B$$

$$m_k = \frac{Y^{(k)}(s_0)}{k!} = (-1)^k C^T (s_0 I - A)^{-(k+1)} B$$

So, adjusting the code as follows.

```
    …..
    moments = zeros(1, num_moments);
    for k = 1:num_moments
        moments(k) = (-1)^(k-1) * C' * (s0 * eye(size(A)) - A)^-(k) * B;
    end
    moments(1)=moments(1)+D;
 …..
```

Which can be tested as follows for the impulse response:

```
clear
clc
% test AWE
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];
D = 0;
t_end = 2;
wo = 10*pi;
[h,y,t]=AWE(A,B,C,D,wo,t_end);
plot(t,h)
```

```
xlabel('time s')
ylabel('H(t)')
title(['Impulse response at w= ', num2str(wo)]);
grid on
```