Implement AWE \_updated

Name: Mohammed AL Shuaili

Date: 27/1/2025

AWE involves 4 main steps:

1. Form a state – space representation

2. Form the moments

3. Find the poles of the system

4. Find the residues

And then form the impulse response as:

(1)

The following code implements AWE with first and second approximation:

1. clear all

2. clc

3.

4. % Input state-space matrices

5. A = [-2 1 0 0; 1 -2 1 0; 0 1 -2 1; 0 0 1 -1];

6. B = [1; 0; 0; 0];

7. C = [1; 0; 0; 0];

8.

9. % Determine the order of the system

10. q = length(B);

11.

12. % Compute moments

13. num\_moments = 2 \* q;

14. moments = zeros(1, num\_moments);

15. for i = 1:num\_moments

16. moments(i) = -transpose(C) \* (A^(-i)) \* B;

17. end

18.

19. % Generalized approximation for all orders

20. for approx\_order = 1:q

21. fprintf('Case %d:\n', approx\_order);

22.

23. % Construct the moment matrix

24. moment\_matrix = zeros(approx\_order);

25. Vector\_c = -moments(approx\_order+1:2\*approx\_order)';

26.

27. for i = 1:approx\_order

28. moment\_matrix(i, :) = moments(i:i+approx\_order-1);

29. end

30. % find b matrix (deno coef)

31. b\_matrix = inv(moment\_matrix)\*Vector\_c;

32.

33. %find the ploes

34. poles = roots([transpose(b\_matrix) ,1]);

35.

36. % determine residuses

37. % form the V matrix

38. V = zeros(approx\_order);

39. for i = 1:approx\_order

40. for j = 1:approx\_order

41. V(i, j) = 1/poles(j)^(i-1);

42. end

43. end

44. % form the A matrix

45. A\_diag = diag(1 ./ poles);

46. r\_moments = moments(1:approx\_order); % a helper matrix

47. % find the residuse

48. residues = -1\*inv(A\_diag)\* inv(V)\* transpose(r\_moments);

49. %set a value for t

50. %t=0:10;

51. t = 0:0.1:5;

52. %form the impulse response

53. h =0;

54. for i = 1:approx\_order

55. h = h + residues(i) \* exp(poles(i) \* t);

56. end

57. % plot the output

58. figure(approx\_order);

59. title(['Apprximation of order',num2str(approx\_order)]);

60. plot(t,h);

61. xlabel('Time (\mus)');

62. ylabel('V Load (Volts)');

63. grid on

64. end

65.

This code takes the input as matrices A, B and C. For example let:

(2)

First one must find the moments as follows:

Where (i) goes from 0 to 2q-1 and q is the order of the transfer function associated with the equation or the state space.

In example 1 we can find that,

Next, find b (coefficients of s in the Laplace expression) as follows:

Then solve for B(s)=0 to obtain the poles of the system where:

Next, finding the residues as:

So, back to example 1, since q = 4, we can find that for a first order approximation:

Hence using the general expression in (1) we obtain,

in the same manner we can then find second, third and fourth order approximations.

* RLC ladder implementation

Now, consider the open voltage RLC ladder for the transmission line with N=2 as follows:

A diagram of a circuit

Description automatically generated

From the circuit, we can say that:

(3)

*So,*

Let,

Rewriting the equations,

Now, express them in terms of the state variables.

1. For

1. For
2. For
3. For

**Step 2: Write in State-Space Form**

Where:

Matrix A:

Matix B:

Matrix C:

Now, let’s implement AWE with these on MATLAB and compare it to ode45 for validation.

This is the expected output (from ode45).

A graph with a line

Description automatically generated

The cod:

1. clear all

2. clc

3. l = 400;

4. N = 2;

5. dz = l/N;

6. R = 0.1\*dz;

7. L = 2.5e-7\*dz;

8. C = 1e-10\*dz;

9. Rs = 0;

10. Vs = 30; % this is u

11. A = [-(Rs+R)/L, -1/L, 0 , 0 ;

12. 1/C , 0 , -1/C, 0 ;

13. 0 , 1/L , -R/L, -1/L;

14. 0 , 0 , 1/C , 0 ];

15. B = [1/L;0;0;0].\*Vs;

16. C = [0;0;0;1];

17. q = length(B);

18. moments = zeros(1,2\*q-1);

19. for i=1:length(moments)

20. moments(i) = -1\*transpose(C)\*A^-i\*B;

21. end

22. %for first order approximation (Case 1)

23. b = -moments(2)/moments(1);

24. %the poles

25. p = -1/b;

26. % residues

27. k=-moments(1)\*p;

28. %t =

29. %hense

30. t = 0:1e-10:20e-6;

31. ht1 = k\*exp(p\*t);

32. %Case 2

33. m2 = [moments(1),moments(2);moments(2),moments(3)];

34. m2\_2 = -1\*[moments(3);moments(4)];

35. b\_case2 = m2^-1\*m2\_2;

36. p\_case2 = roots([b\_case2(1),b\_case2(2),1]);

37. %residues

38. V = [1 1 ;1/p\_case2(1) 1/p\_case2(2)];

39. A\_case2 = [1/p\_case2(1) 0;0 1/p\_case2(2)];

40. k\_case2 = -1\* inv(A\_case2) \* inv(V) \* [moments(1);moments(2)];

41. % hence the final expersion is

42. ht\_case2 = k\_case2(1)\*exp(p\_case2(1)\*t)+k\_case2(2)\*exp(p\_case2(2)\*t);

43. figure(1)

44. plot(t,ht1);

45. grid on

46. xlabel('Time (\mus)');

47. ylabel('V Load (Volts)');

48. title(['RLC with AWE first order at N = ',num2str(N)]);

49. figure(2)

50. plot(t,ht\_case2);

51. grid on

52. xlabel('Time (\mus)');

53. ylabel('V Load (Volts)');

54. title(['RLC with AWE second order at N = ',num2str(N)]);

55.

A screenshot of a graph

Description automatically generated

Looking at the first and second order approximations, they don’t match the expected outputs from ode45.

Let’s investigate the third order approximation.

Using the matrices above, the moments can be found as follows:

Thus,

And ,

Hence,

A graph with a line

Description automatically generated

This is still not close to the expected result as in ode45.

Now, Let’s consider the following circuit (RLC ladder with N = 3 and open voltage) to find a pattern where we can link the number of sections to the state space model:

A diagram of a circuit

Description automatically generated

Let,

So, each section of the transmission line contributes two states to the state-space model:

What to do next (in the meantime):

1. Find what causing the issue for the AWE

Y and S parameters:

Y-parameters, or admittance parameters, characterize the electrical behavior of transmission lines in network analysis by relating port voltages and currents. They are used for modeling high-frequency circuits, especially in microwave design, as they describe how a network admits current in response to voltage.

For a two-port network:

* **Y11**: Input admittance at port 1 (port 2 shorted).
* **Y12**: Reverse admittance from port 2 to port 1.
* **Y21**: Forward admittance from port 1 to port 2.
* **Y22**: Output admittance at port 2 (port 1 shorted).

These parameters are key for understanding transmission line interactions with other components [1].

[1] A. Arrais and R. Levy, “Direct Y-Parameter Estimation of Microwave Structures Using TLM Simulation and Prony's Method,” *ResearchGate*, [Online]. Available: <https://www.researchgate.net/publication/230844927_Direct_Y-parameter_estimation_of_microwave_structures_using_TLM_simulation_and_Prony's_method>. [Accessed: Jan. 25, 2025].

Further research can be done, at this time the IEEE website is down.

A blue background with white text

Description automatically generated