CFH Test

Name: Mohammed Alshuaili

Date : 3/12/2025

Consider the following poles:

The transfer function is then,

Testing AWE with w=0 and w = 2.5j, we get the correct poles and residues:

Code

clear

clc

num=[6,26,48,32] ; %results from the given exxample

deno = [1 6 18 24 16];

[A,B,C,D] = create\_state\_space(num,deno);

w = 2.5i;% expnation point

[p,~,r] = AWE\_poles(A,B,C,D,w);

 p

looking at the transmission line with the following parameters:

2 models:

|  |  |  |
| --- | --- | --- |
| w | Poles | Residues |
| AWE\_, Y\_W(1:10) | -1.5223 + 7.3975i  -1.5223 - 7.3975ix | -0.7784 - 4.0332i  -0.7784 + 4.0332ix |
| AWE\_Y\_W(11:15) | -0.3748 + 7.7129i  -0.3762 - 7.7037i | 2.5440 - 0.2574i  2.5410 + 0.2553i |

The results are almost the same as the first model

Code:

clear

clc

% Generate frequency points

f = linspace(0, 9e5, 100);

w = 2\*pi\*f;

s = 1i \* w;

t = 50e-6;

first\_idx = 1:10;

vo =1./(cosh(400.\*(1e-10.\*s).^(1/2).\*(0.1+2.5e-7.\*s).^(1/2)));

[H1a,num,deno] = generate\_yp2(real(vo(first\_idx)),imag(vo(first\_idx)),w(first\_idx));

[A,B,C,D] = create\_state\_space(num,deno);

[~,H1,~,~,p1,m1]=AWE2(A,B,C,D,0,30,t);% to compare it with the results

[p1,np1,r1,m1] = AWE\_poles(A,B,C,D,w(first\_idx(1)));

%second model ----------------------------------------------------------

idx = 11:15; % f = 9.09e4 : 1.2727e+05

H\_diff = vo(idx)-H1a(s(idx));

[H2a,num,deno] = generate\_yp2(real(H\_diff),imag(H\_diff),w(idx));

%[H2a,num,deno] = generate\_yp2(real(vo(idx)),imag(vo(idx)),w(idx));

[A,B,C,D] = create\_state\_space(num,deno);

[p2,np2,r2,m2] = AWE\_poles(A,B,C,D,150);

test\_poles = [p1',p2'];

% remove unstable poles

for i=1:length(test\_poles)

if real(test\_poles(i))<0

ptest(i) = test\_poles(i);

end

end

moments = m1; % moments from the first model has 1 value and zeros

[hs,r]= generate\_hs(ptest,length(ptest),moments);

RMSE\_idx = 1:25;

R1 = RMSE(hs(s(RMSE\_idx)),vo(RMSE\_idx));

R2 = RMSE(H1a(s(RMSE\_idx)),vo(RMSE\_idx));

plot(f,hs(s),f,vo,f,H1a(s),'ro');

legend('result','exact','first model');

xlabel('f (Hz)')

AWE\_poles :

function [poles,poles\_unshifted,residues,moments]= AWE\_poles(A, B, C, D, w)

q = length(B);

num\_moments = 2 \* q;

s0 = 1i \* w;

moments = zeros(1, num\_moments);

[r, c] = size(C);

if r ~= 1

C = C';

end

for k = 1:num\_moments

moments(k) = (-1)^(k-1)\*C\*(s0\*eye(size(A))- A)^-(k)\*B;

end

moments(1) = moments(1) + D; % Include D in the zeroth moment

approx\_order = q;

% Construct moment matrix and vector for denominator coefficients

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 = moment\_matrix \ Vector\_c;

poles\_unshifted = roots([b\_matrix; 1]); % Unshifted poles (s' = s - s0)

% Compute residues using unshifted poles

V = zeros(approx\_order);

for i = 1:approx\_order

for j = 1:approx\_order

V(i, j) = 1 / (poles\_unshifted(j))^(i-1);

end

end

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

r\_moments = moments(1:approx\_order);

residues = -A\_diag \ (V \ r\_moments(:));

% Shift poles to s-plane

poles = poles\_unshifted+s0;

end

generate hs

function [h\_s,residues]= generate\_hs(poles,q,moments)

approx\_order =q;

% Compute residues using given poles and moments

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 = -A\_diag \ (V \ r\_moments(:));

% Transfer function in s-domain

h\_s = @(s)0;

for i =1:length(poles)

h\_s = @(s) h\_s(s)+residues(i) ./ (s - poles(i));

end

end