W8

Name: Mohammed AL Shuaili

Date: 2/3/2025

* Complex frequency

Consider 4 models as follows using AWE\_CHF with M = 6, where

|  |  |
| --- | --- |
| w | Poles |
| AWE\_, Y\_W(1:20) | 1.6906 + 0.0000i  -1.1379 + 0.0000i  0.3416 + 0.7967i  0.3416 - 0.7967i  -0.1938 + 0.8052i  -0.1938 - 0.8052i |
| AWE\_, Y\_W(21:30) | -4.1884 + 0.2317i  1.9618 + 3.1320i  -1.3208 - 0.0000i  -0.4097 + 0.0000i  0.2323 + 1.8643i  -0.0000 + 1.1424i |
| AWE\_, Y\_W(31:40) | -0.2437 - 2.3025i  0.5619 + 2.0314i  -0.2437 + 2.3025i  -0.6176 + 1.3746i  0.0937 + 1.4062i  -0.0000 + 1.7136i |
| AWE\_, Y\_W(41:50) | -0.2194 - 2.3773i  -0.2194 + 2.3773i  0.0847 + 2.4142i  0.0792 + 2.1397i  -0.0595 + 2.1806i  0.0000 + 2.2848i |

1. Consider the first 3 model and remove all unstable poles we get:

|  |
| --- |
| -0.2437 - 2.3025i  -0.2437 + 2.3025i  -0.6176 + 1.3746i  -0.0000 + 1.7136i  -4.1884 + 0.2317i  -1.3208 - 0.0000i  -0.4097 + 0.0000i  -0.0000 + 1.1424i  -1.1379 + 0.0000i  -0.1938 + 0.8052i  -0.1938 - 0.8052i |

Since there is no overlapping, we assume all these poles are valid and dominant.

1. Find the residues,

We first need to determine the moments,

|  |  |
| --- | --- |
| w | moments |
| AWE\_, Y\_W(1:20) | 1.0874, , ,…. |
| AWE\_, Y\_W(21:30) | -1.1042 - 0.4466i, ,… |
| AWE\_, Y\_W(31:40) | -0.8856 + 0.1380i, |

Using the moments of the first model, generated residues are:

|  |
| --- |
|  |

All of these residues are close to 0 except for the one highlighted and these are the one associated with the poles from the first model. Hence, the final model is almost the same as the first model. The resultant mode highly depends on the moments’ matrix.

This is not automated and not good implementation, it’s just to obtain an understanding before automating it.

clear

clc

% Generate frequency points

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

w = 2\*pi\*f;

s = 1i \* w;

M = 6;

t = 50e-6;

first\_idx = 1:20;

vo =1./(cosh(400.\*(0 + 1e-10.\*s).^(1/2).\*(0.1 + 2.5e-7.\*s).^(1/2)));

[H1,num,deno] = generate\_yp2(real(vo(first\_idx)),imag(vo(first\_idx)),w(first\_idx));

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

[p1c,np1c,r1c,m1c] = AWE\_CFH\_poles(A,B,C,D,M,w(first\_idx(1)));

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

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

idx = 21:30;

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

[H2,num,deno] = generate\_yp2(real(vo(idx)),imag(vo(idx)),w(idx));

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

[p2c,np2c,r2c,m2c] = AWE\_CFH\_poles(A,B,C,D,M,w(idx(1)));

[p2,np2,r2,m2] = AWE\_poles(A,B,C,D,w(idx(1)));

% Third model ---------------------------------------------

idx = 31:40;

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

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

[H3,num,deno] = generate\_yp2(real(vo(idx)),imag(vo(idx)),w(idx));

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

[p3c,np3c,r3c,m3c] = AWE\_CFH\_poles(A,B,C,D,M,w(idx(1)));

[p3,np3,r3,m3] = AWE\_poles(A,B,C,D,w(idx(1)));

% Forth model ---------------------------------------------------

idx = 41:50;

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

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

[H4,num,deno] = generate\_yp2(real(vo(idx)),imag(vo(idx)),w(idx));

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

[p4c,np4c,r4c,m4c] = AWE\_CFH\_poles(A,B,C,D,M,w(idx(1)));

[p4,np4,r4,m4] = AWE\_poles(A,B,C,D,w(idx(1)));

poles\_c = [p1c,p2c,p3c,p4c];% poles from AWE\_CFH with many moments

poles\_nc = [np1c,np2c,np3c,np4c];% shifted poles

poles = [p1,p2,p3,p4]; % AWE with q = lenght(B)

polesn = [np1,np2,np3,np4]; %shifted poles

pt = [p3c',p2c',p1c'];

ptest = 0;

% remove unstable poles

for i=1:length(pt)

if real(pt(i))<0

ptest = [ptest,pt(i)];

end

end

ptest = ptest(2:end);

mtest = m1c; %% moments from the first model has 1 value and zeros

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

%plot(f,hs(s),f,vo,f,H1(s),'r\*');

A graph of a function

AI-generated content may be incorrect.

Figure 1: first model Vs resultant model

Case 2, largest residues:

1. Increase the dimensions of the A matrix to 3x3, currently it’s a 2x2 matrix.

Rounding issue with tol = 8.589176e-02., so these results might not be accurate.

|  |  |  |
| --- | --- | --- |
| w | Poles | Residues |
| AWE\_, Y\_W(1:20) | -2.2814 + 0.0000i  -0.2105 + 0.7606i  -0.2105 - 0.7606i | -8.8177 - 0.0000i  0.0032 - 5.6259i  0.0032 + 5.6259i |
| AWE\_, Y\_W(21:30) | -4.1540 - 0.4381i  0.2075 + 0.9514i  -0.0003 + 1.1422i | -4.8632 - 0.9176i  0.1442 + 0.1181i  -0.0000 - 0.0000i |
| AWE\_, Y\_W(31:40) | -0.6004 - 0.0610i  -0.2231 + 2.3680i  -0.0065 + 1.7156i | 3.1082 - 3.9632i  0.3882 + 5.7094i  -0.0000 - 0.0000i |
| AWE\_, Y\_W(41:50) | -0.5261 - 0.0711i  -0.2396 + 2.3472i  -0.0000 + 2.2849i | 6.4974 - 4.9537i  0.4054 + 6.5539i  0.0000 + 0.0000i |

Considering the first 2 models and removing unstable poles, then considering the one with the largest residue we get:

|  |
| --- |
| -0.2105 - 0.7606i, -0.0003 + 1.1422i |

A graph of a function

AI-generated content may be incorrect.A graph of a function

AI-generated content may be incorrect.

Figure 2 :impulse response of the first model Vs exact model Vs resultant model (unshifted vs shifted poles).

It can be seen that the resultant model is not more accurate than the first model which we want to achieve.

Consider model 3 and 4.

|  |
| --- |
| -0.6004 - 0.0610i ,-0.5261 - 0.0711i |

A graph of a function

AI-generated content may be incorrect.A graph of a graph

AI-generated content may be incorrect.

Figure 3: impulse response of the exact model Vs resultant model (unshifted vs shifted poles).

The same as in the previous case, now consider more poles from each model, say 2 poles from each model.

the first 2 models.

|  |
| --- |
| -0.2105 - 0.7606i, -0.2105 + 0.7606i, -0.0003 + 1.1422i |

A graph of a graph

AI-generated content may be incorrect.A graph of a function

AI-generated content may be incorrect.

Figure 4: impulse response of the resultant model Vs the exact model (shifted and unshifted).

With RMSE 0.3972 compared to 0.0165 of the first model.

## Now, let’s consider expansion points near the imaginary part of the poles of the first model (0.7606):

|  |  |  |
| --- | --- | --- |
| w | Poles | Residues |
| AWE\_, Y\_W(1:20) | -2.2814 + 0.0000i  -0.2105 + 0.7606i  -0.2105 - 0.7606i | -8.8177 - 0.0000i  0.0032 - 5.6259i  0.0032 + 5.6259i |
| AWE\_, Y\_W(14:30) | -1.6442 + 0.8550i  -0.1184 + 0.6931i  0.0001 + 0.7425i | -1.3628 + 0.3149i  0.3148 - 0.2773i  0.0000 - 0.0000i |

Considering the following poles:

|  |
| --- |
| -0.2105 - 0.7606i, -0.2105 + 0.7606i, -1.6442 - 0.1124i, -0.1184 + 0.0495i |

A graph of a function

AI-generated content may be incorrect.A graph of a function

AI-generated content may be incorrect.

Figure 5 : shows the impulse response of the exact solution compared to resultant model (shifted and unshifted)

With RMSE 0.0507 or 0.0412 for shifted and unshifted poles respectively.

## Now, let’s consider evaluating the second model as the difference between the exact model and the first model at the range of frequencies.

|  |  |  |
| --- | --- | --- |
| w | Poles | Residues |
| AWE\_, Y\_W(1:20) | -2.2814 + 0.0000i  -0.2105 + 0.7606i  -0.2105 - 0.7606i | -8.8177 - 0.0000i  0.0032 - 5.6259i  0.0032 + 5.6259i |
| AWE\_, Y\_W(21:30) | -0.0746 - 0.1820i  -0.2505 + 1.6580i  0.0014 + 1.1530i | -1.2972 + 0.0725i  -0.7350 - 0.0402i  -0.0000 - 0.0000i |
| AWE\_, Y\_W(31:40) | -0.7546 - 0.0160i  -0.1737 + 2.2948i  -0.0036 + 1.7130i | 1.0823 - 2.3855i  -0.5038 + 3.6336i  0.0000 - 0.0000i |
| AWE\_, Y\_W(41:50) | -0.2502 - 0.0849i  -0.2343 + 2.4916i  0.0000 + 2.2847ix | 4.5658 - 1.4445i  0.7652 + 1.9364i  -0.0000 - 0.0000i |

Consider the first 2 models with the largest residue, we get

|  |
| --- |
| -0.2105 - 0.7606i, -0.2105 + 0.7606i, -0.2505 - 0.5156i |

A graph of a function

AI-generated content may be incorrect.

Figure 6 : shows the exact response compared to the result.

With RMSE 0.1560,

Try using the moments of the second model, we get. A graph of a wave

AI-generated content may be incorrect.

## Now, let’s consider the poles and their residues.

A graph of a function

AI-generated content may be incorrect.

Figure 7: shows the sum of poles and their residues.

The following code is used for testing purpose only:

%%

clear

clc

% Generate frequency points

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

w = 2\*pi\*f;

s = 1i \* w;

M = 6;

t = 50e-6;

first\_idx = 1:20;

vo =1./(cosh(400.\*(0 + 1e-10.\*s).^(1/2).\*(0.1 + 2.5e-7.\*s).^(1/2)));

[H1a,num,deno] = generate\_yp(real(vo(first\_idx)),imag(vo(first\_idx)),w(first\_idx)); %3 order TF

[A,B,C,D] = create\_state\_space(num,deno); % will generate 3x3 A matrix

[~,H1]=AWE2(A,B,C,D,w(1),30,t); % this is for comparison

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

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

idx = 14:30;

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

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

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

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

[~,H2]=AWE2(A,B,C,D,w(idx(1)),30,t);

[p2,np2,r2,m2] = AWE\_poles(A,B,C,D,w(idx(1)));

% Third model ---------------------------------------------

idx = 31:40;

%H\_diff = vo(idx)-H2a(s(idx));

[H3s,num,deno] = generate\_yp(real(vo(idx)),imag(vo(idx)),w(idx));

%[H3s,num,deno] = generate\_yp(real(H\_diff),imag(H\_diff),w(idx));

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

[~,H3]=AWE2(A,B,C,D,w(idx(1)),30,t);

[p3,np3,r3,m3] = AWE\_poles(A,B,C,D,w(idx(1)));

% Forth model ---------------------------------------------------

idx = 41:50;

%H\_diff = vo(idx)-H3s(s(idx));

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

[H4a,num,deno] = generate\_yp(real(vo(idx)),imag(vo(idx)),w(idx));

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

[p4,np4,r4,m4] = AWE\_poles(A,B,C,D,w(idx(1)));

[~,H4]=AWE2(A,B,C,D,w(idx(1)),30,t);

poles = [p1,p2,p3,p4]; % AWE with q = lenght(B)

polesn = [np1,np2,np3,np4]; %shifted poles

pt = [p1',p2'];

rt = [r1',r2'];

ptest = 0;

rtest = 0;

% remove unstable poles

for i=1:length(pt)

if real(pt(i))<0

ptest = [ptest,pt(i)];

rtest = [rtest,rt(i)];

end

end

%ptest = [ptest(3:4),ptest(end)];

ptest = ptest(2:end);

rtest = rtest(2:end);

%ptest = [ptest(2:3),ptest(end)];

mtest = [m1,m2]; %% moments from the first model has 1 value and zeros

[hs,r]= generate\_hs(ptest,length(ptest),mtest,w(35));

%hs = @(s) hs(s)+H1(s);

RMSE\_idx = 1:20;

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

R2 = RMSE(H1(s(RMSE\_idx)),vo(RMSE\_idx),length(vo(RMSE\_idx)));

plot(f,hs(s),f,vo);

legend('result','exact');

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,w)

s0=1i\*w;

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

% this for testing only

poles = poles+s0;

hs = @(s)0;

for i =1:length(poles)

hs = @(s) hs(s)+residues(i) ./ ((s) - poles(i));

end

end