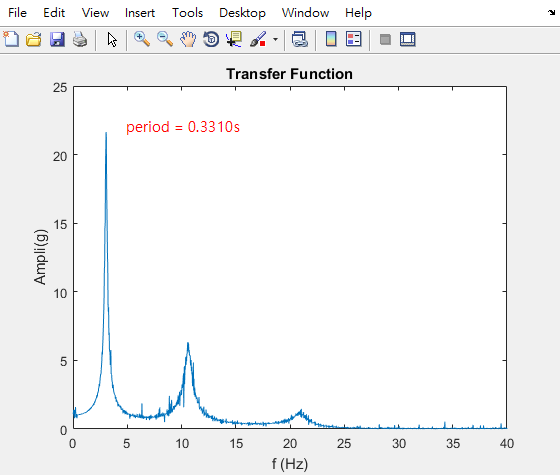
結構控制 HW1

1. 依據題目給定的條件，尋找結構週期。



得到題目給的第一模態的週期為：0.3310s

clc; clear; close all;

El50AXBS\_fileid = fopen('El50AXBS.txt', 'r');

EL50RFA\_fileid = fopen('EL50RFA.txt', 'r');

El50AXBS = fscanf(El50AXBS\_fileid, '%f');

EL50RFA = fscanf(EL50RFA\_fileid, '%f');

[EL50RFA\_freq, EL50RFA\_ampl] = fft\_improve(EL50RFA, 0.0125);

[El50AXBS\_freq, El50AXBS\_ampl] = fft\_improve(El50AXBS, 0.0125);

transfer\_function = EL50RFA\_ampl ./ El50AXBS\_ampl;

plot(El50AXBS\_freq(2 : end), transfer\_function(2 : end));

[transfer\_function\_max, transfer\_function\_index]=max(transfer\_function(2 : end));

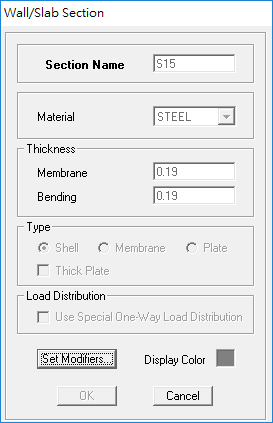
1 / El50AXBS\_freq(transfer\_function\_index)

title('Transfer Function');

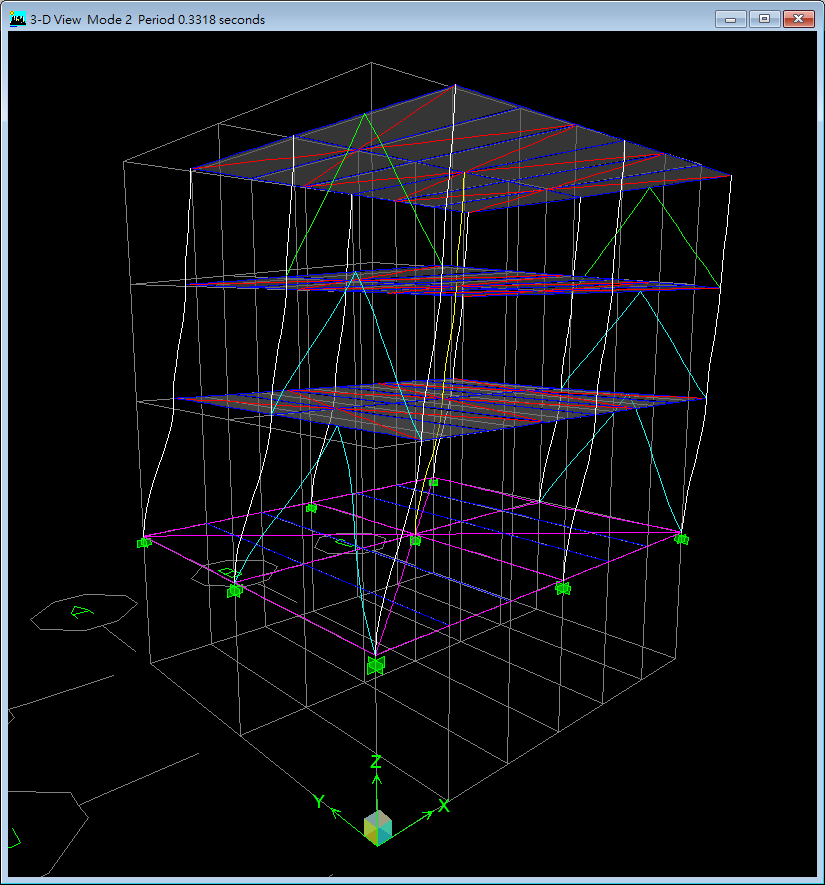
xlabel('f (Hz)');

ylabel('Ampli(g)');

1. 找到給定的結構週期之後，調整版厚來對應到題目給的週期。

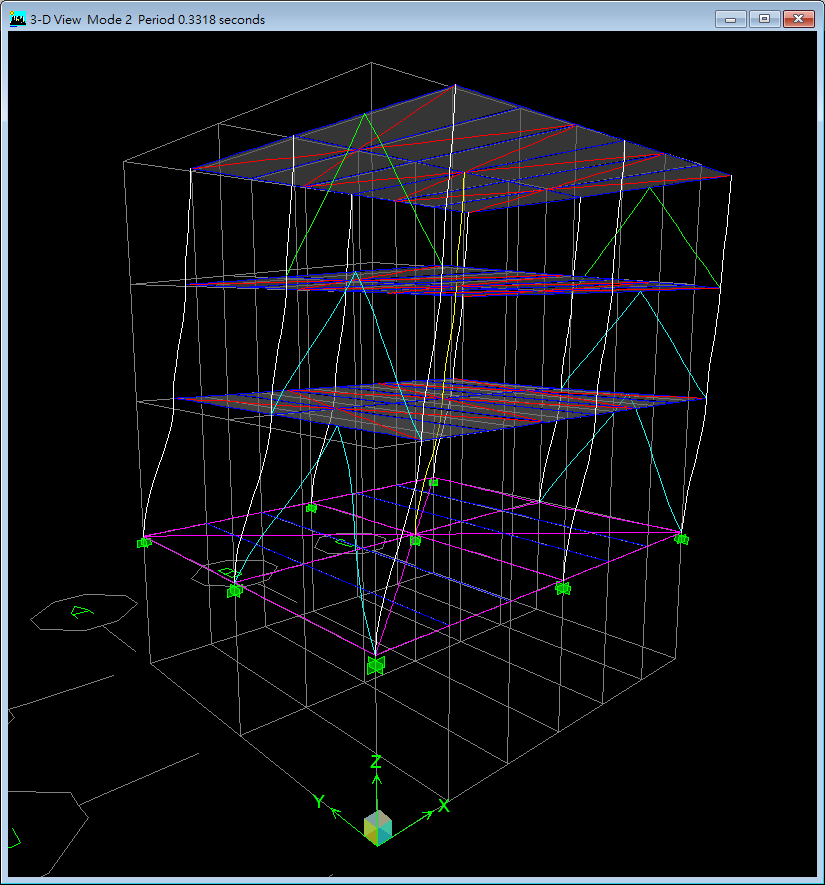
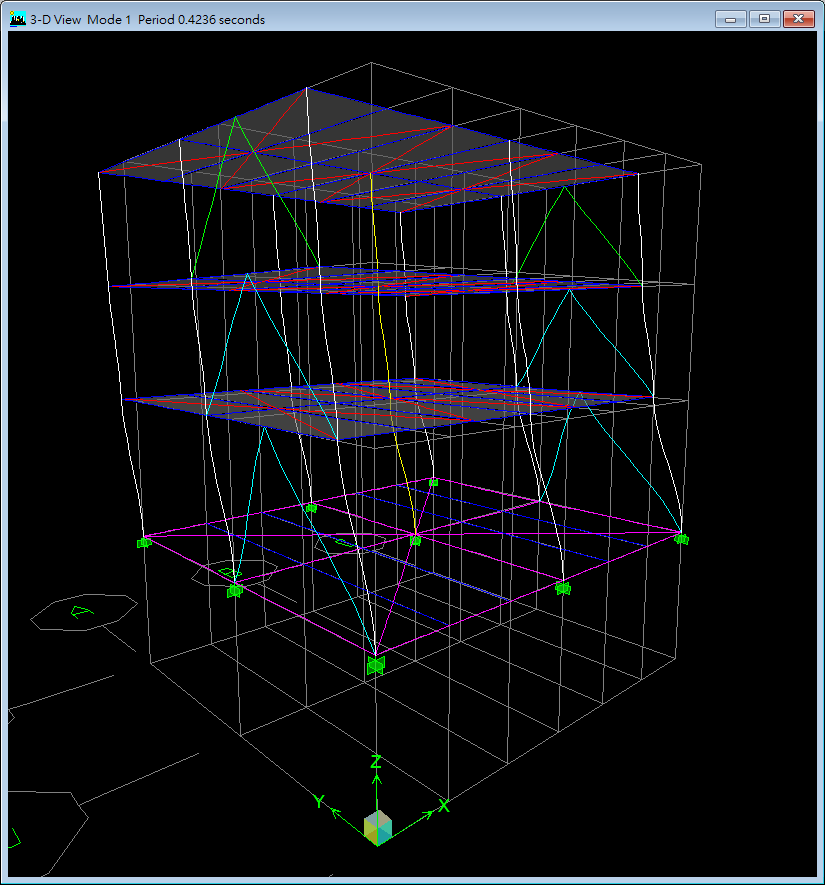


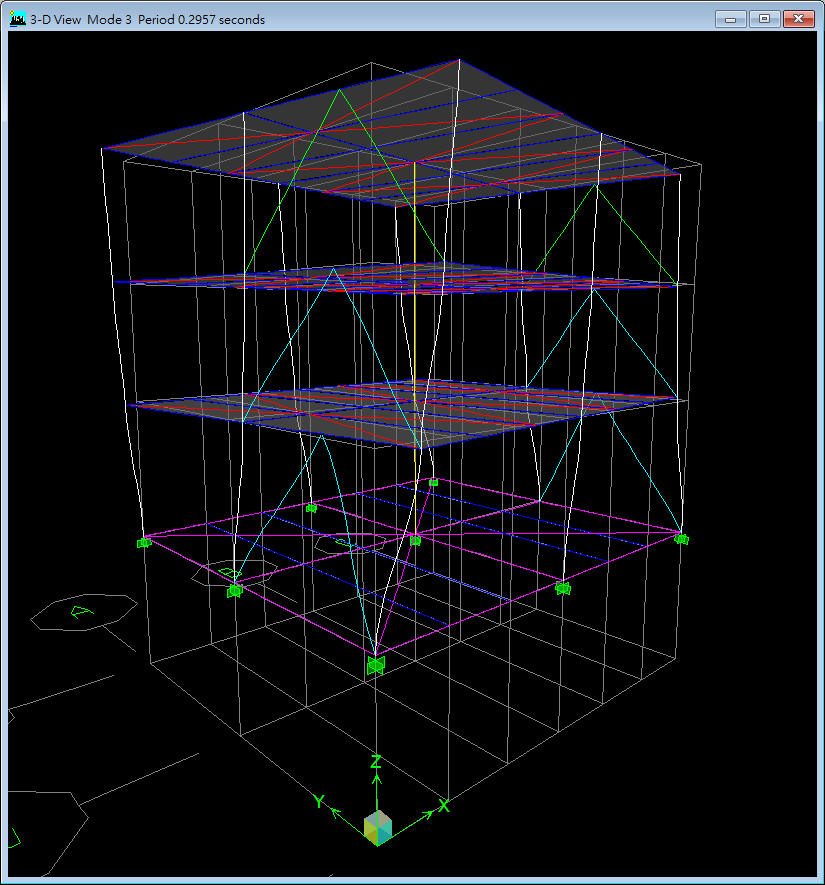
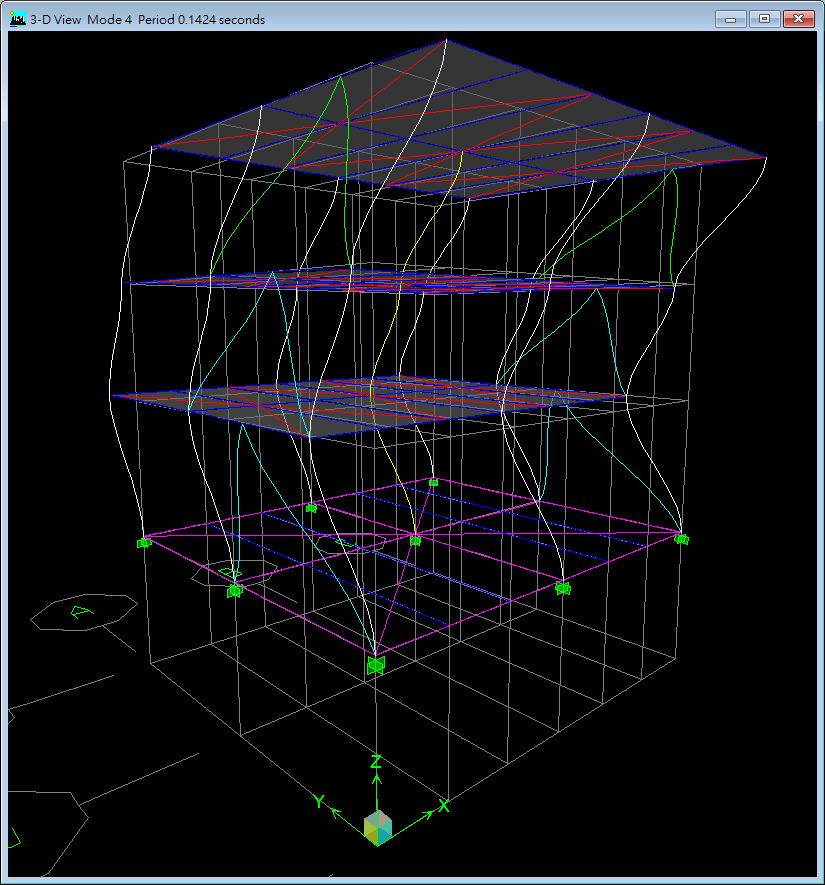
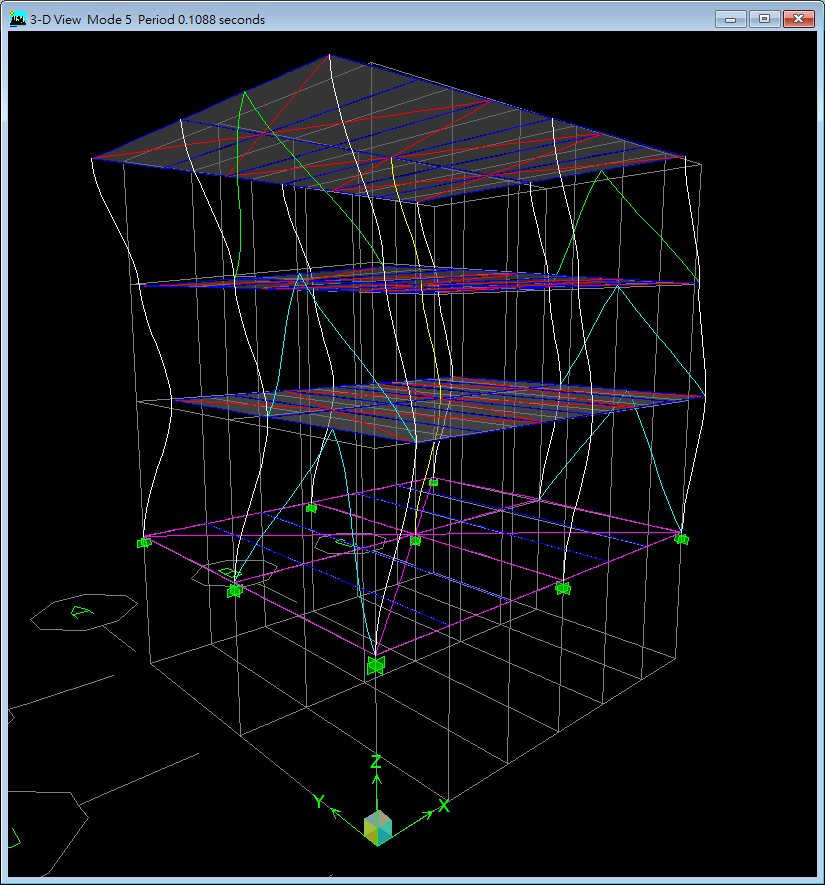
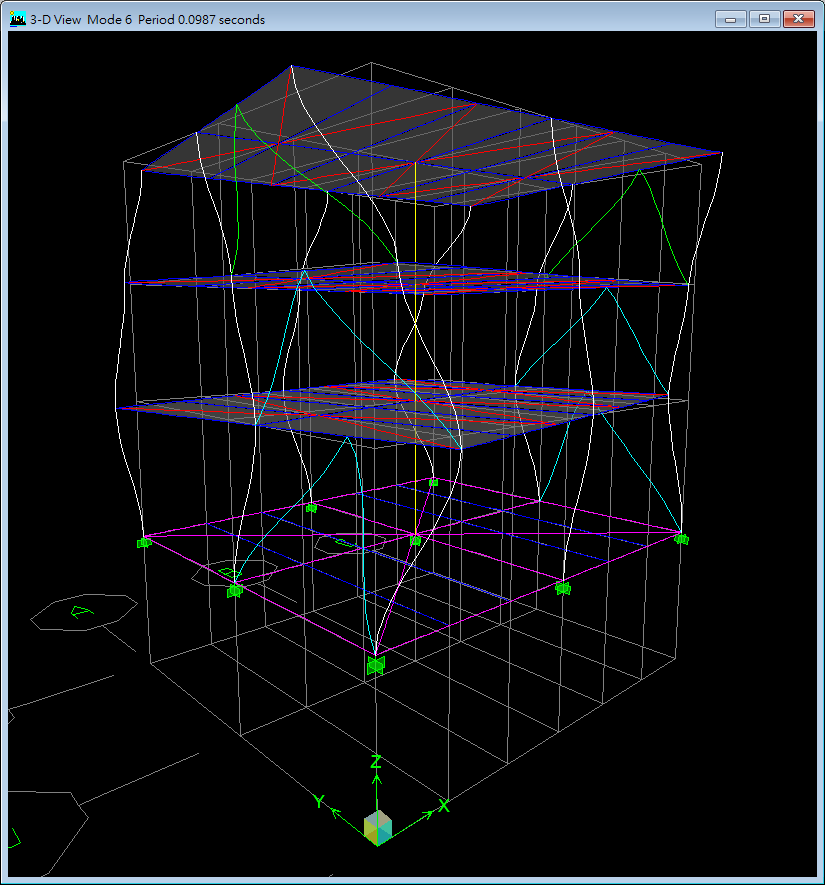
版厚調整為 19 公分。



模型週期為：0.3318s

6 個 mode shapes

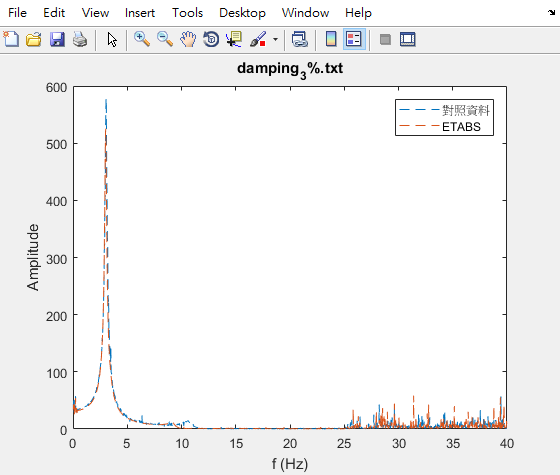


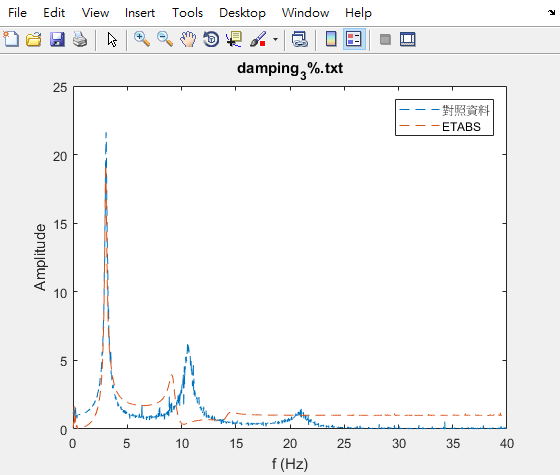
1. 調整完成週期之後，調整模型的阻尼比來對應到題目給的阻尼比。

3%

樓頂位移 Trensfer Function

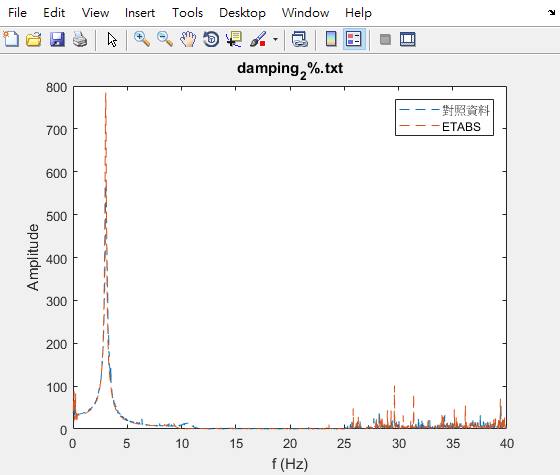


樓頂加速度 Trensfer Function

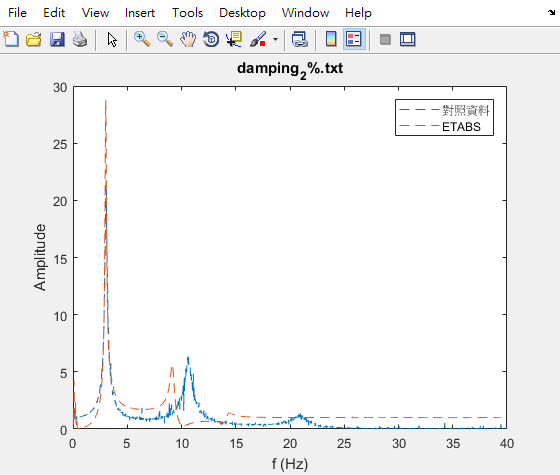


2%

樓頂位移 Trensfer Function

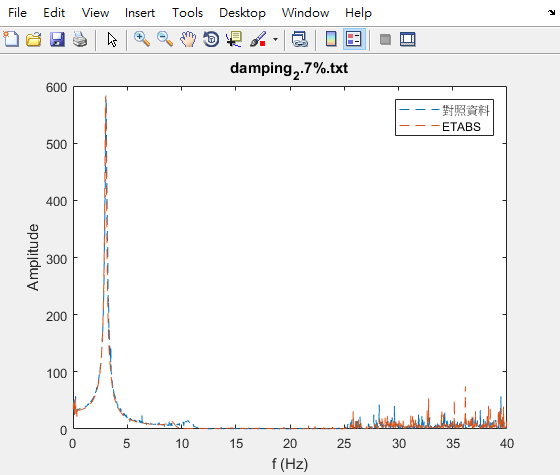


樓頂加速度 Trensfer Function

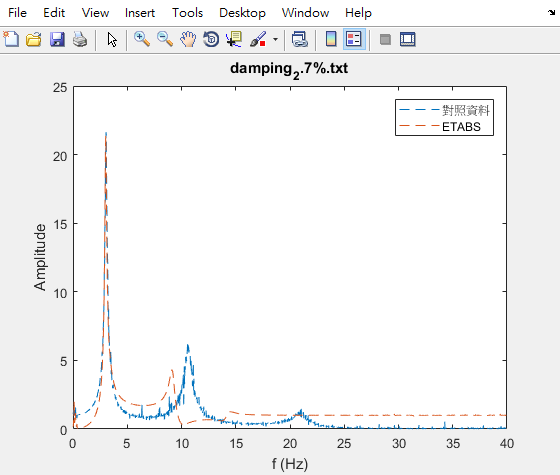


2.7%

樓頂位移 Trensfer Function



樓頂加速度 Trensfer Function



多次試誤後得到 2.7% 的阻尼比最吻合題目給的資料。

function [] = find\_damping(*file*)

TIME\_COL = 1;

DISP\_COL = 2;

ACCEL\_COL = 3;

BASE\_COL = 4;

El50AXBS\_fileid = fopen('El50AXBS.txt', 'r');

El50AXBS = fscanf(El50AXBS\_fileid, '%f');

EL50RFA\_fileid = fopen('EL50RFA.txt', 'r');

EL50RFA = fscanf(EL50RFA\_fileid, '%f');

EL50RFD\_fileid = fopen('EL50RFD.txt', 'r');

EL50RFD = fscanf(EL50RFD\_fileid, '%f');

damping\_fileid = fopen(file, 'r');

damping = fscanf(damping\_fileid, '%f %f %f %f', [4 Inf]);

damping = damping';

damping(:, DISP\_COL) = damping(:, DISP\_COL) \* 1000;

damping(:, ACCEL\_COL) = damping(:, ACCEL\_COL) / 9.81;

damping(:, BASE\_COL) = damping(:, BASE\_COL) / 9.81;

figure;

plot((1 : length(EL50RFA)) \* 0.0125, EL50RFA, damping(:, TIME\_COL), damping(:, ACCEL\_COL), '--')

figure;

[~, El50AXBS\_ampl] = fft\_improve(El50AXBS, 0.0125);

[EL50RFA\_freq, EL50RFA\_ampl] = fft\_improve(EL50RFA, 0.0125);

input\_transfer\_function = EL50RFA\_ampl ./ El50AXBS\_ampl;

plot(EL50RFA\_freq(2 : end), input\_transfer\_function(2 : end), '--');

hold on;

[damping\_freq, damping\_ampl] = fft\_improve(damping(:, ACCEL\_COL), 0.0125);

[~, base\_ampl] = fft\_improve(damping(:, BASE\_COL), 0.0125);

output\_transfer\_function = damping\_ampl ./ base\_ampl;

plot(damping\_freq(2 : end), output\_transfer\_function(2 : end), '--');

title(file);

xlabel('f (Hz)');

ylabel('Amplitude');

legend('對照資料', 'ETABS');

figure;

[~, El50AXBS\_ampl] = fft\_improve(El50AXBS, 0.0125);

[EL50RFD\_freq, EL50RFD\_ampl] = fft\_improve(EL50RFD, 0.0125);

input\_transfer\_function = EL50RFD\_ampl ./ El50AXBS\_ampl;

plot(EL50RFD\_freq(2 : end), input\_transfer\_function(2 : end), '--');

hold on;

[damping\_freq, damping\_ampl] = fft\_improve(damping(:, DISP\_COL), 0.0125);

[~, base\_ampl] = fft\_improve(damping(:, BASE\_COL), 0.0125);

output\_transfer\_function = damping\_ampl ./ base\_ampl;

plot(damping\_freq(2 : end), output\_transfer\_function(2 : end), '--');

title(file);

xlabel('f (Hz)');

ylabel('Amplitude');

legend('對照資料', 'ETABS');

end

clc; clear; close all;

find\_damping('damping\_2.7%.txt');

*% find\_damping('damping\_2.5%.txt');*

find\_damping('damping\_2%.txt');

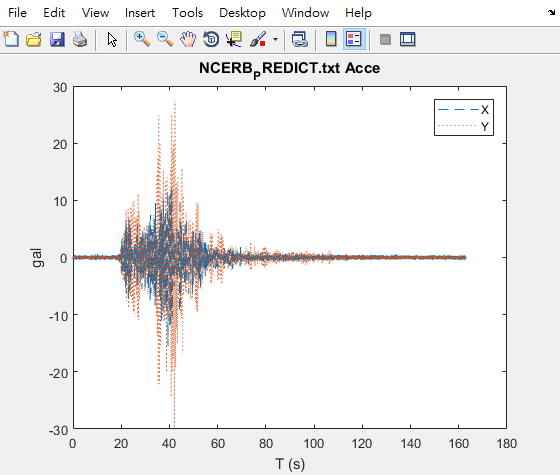
find\_damping('damping\_3%.txt');

*% find\_damping('damping\_2.8%.txt');*

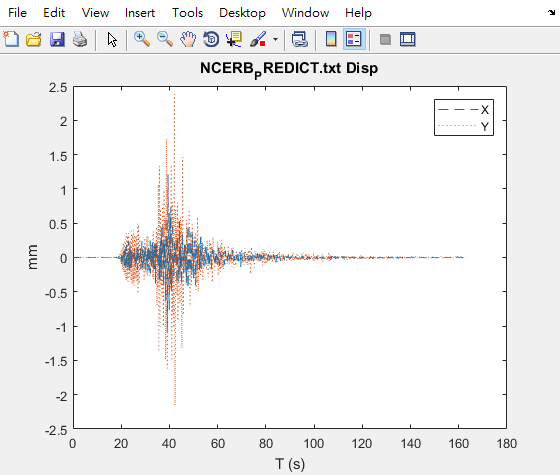
1. 調整完成模型後，進行預測。

NCERB

加速度

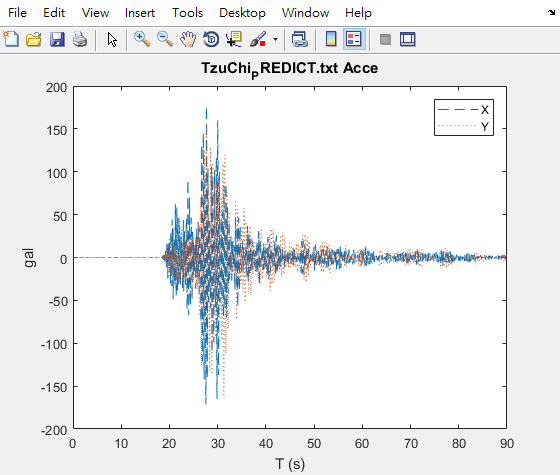


位移

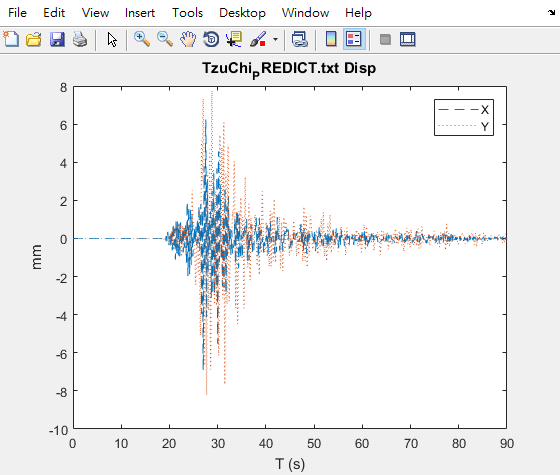


TzuChi

加速度



位移



function [] = predict\_model(*file*)

TIME\_COL = 1;

DISP\_X\_COL = 2;

ACCEL\_X\_COL = 3;

DISP\_Y\_COL = 4;

ACCEL\_Y\_COL = 5;

damping\_fileid = fopen(file, 'r');

damping = fscanf(damping\_fileid, '%f %f %f %f %f', [5 Inf]);

damping = damping';

damping(:, [DISP\_X\_COL, DISP\_Y\_COL]) = damping(:, [DISP\_X\_COL, DISP\_Y\_COL]) \* 10;

figure;

plot(damping(:, TIME\_COL), damping(:, DISP\_X\_COL), '--', damping(:, TIME\_COL), damping(:, DISP\_Y\_COL), ':');

title([file ' Disp']);

xlabel('T (s)');

ylabel('mm');

legend('X', 'Y');

figure;

plot(damping(:, TIME\_COL), damping(:, ACCEL\_X\_COL), '--', damping(:, TIME\_COL), damping(:, ACCEL\_Y\_COL), ':');

title([file ' Acce']);

xlabel('T (s)');

ylabel('gal');

legend('X', 'Y');

end

clc; clear; close all;

predict\_model('NCERB\_PREDICT.txt');

predict\_model('TzuChi\_PREDICT.txt');

1. Matlab Function Code

function [*frequency*, *amplitude*] = fft\_improve(*data*, *time\_interval*)

data\_length = length(data);

data\_fft = fft(data);

data\_abs = abs(data\_fft / data\_length);

data\_half = data\_abs(1 : fix(data\_length / 2 + 1));

data\_half(2 : end - 1) = 2 \* data\_half(2 : end - 1);

amplitude = data\_half;

frequency = 1 / time\_interval \* (0 : fix(data\_length / 2) ) / data\_length;

end