



clc; clear; close all;

ag = filename\_to\_array('I-ELC270\_gal\_l00Hz', 2, 2);

tn = 0.01 : 0.01 : 5;

tn\_length = length(tn);

acceleration = zeros(1, tn\_length);

for index = 1 : tn\_length

[~, ~, a\_array] = newmark\_beta(ag, 0.01, 0.05, tn(index), 'average');

acceleration(1, index) = max(abs(a\_array));

end

acceleration\_normal = acceleration / acceleration(1, 1) \* 0.4;

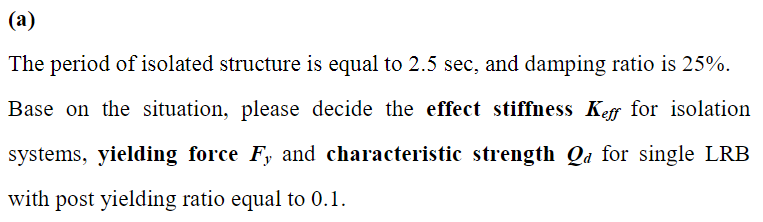
acceleration\_normal(tn == 2.5)

figure;

plot(tn, acceleration\_normal);

xlabel('T(sec)');

ylabel('SaD(g)');



已知：

* TeD = 2.5sec
* Damping Ratio = 25%
* Post Yielding Ratio = 0.1



由反應譜可知 SaD

* TeD => SaD = 0.2324
* W = 35.53421364 (from ETABS)

由以上資訊經迭代可得

* Keff for isolation systems = 22.8801 tf/m
* Fy for single LRB = 0.2791 tf
* Qd for single LRB = 0.2512 tf

clc; clear; close all;

SaD = 0.2324;

TeD = 2.5;

DampingRatio = 0.25;

*% SaD = 0.195;*

*% TeD = 3;*

*% DampingRatio = 0.3;*

W = 35.53421364;

postYieldingRatio = 0.1;

LRBs = 9;

dleta = 0.000001;

g = 9.81;

KeD = W / ((TeD / (2 \* pi)) ^ 2) / g;

if DampingRatio <= 0.02

BS = 0.8;

B1 = 0.8;

elseif 0.02 < DampingRatio && DampingRatio <= 0.05

BS = (DampingRatio - 0.02) / 0.03 \* (1 - 0.8) + 0.8;

B1 = (DampingRatio - 0.02) / 0.03 \* (1 - 0.8) + 0.8;

elseif 0.05 < DampingRatio && DampingRatio <= 0.1

BS = (DampingRatio - 0.05) / 0.05 \* (1.33 - 1) + 1;

B1 = (DampingRatio - 0.05) / 0.05 \* (1.25 - 1) + 1;

elseif 0.1 < DampingRatio && DampingRatio <= 0.2

BS = (DampingRatio - 0.1) / 0.1 \* (1.6 - 1.33) + 1.33;

B1 = (DampingRatio - 0.1) / 0.1 \* (1.5 - 1.25) + 1.25;

elseif 0.2 < DampingRatio && DampingRatio <= 0.3

BS = (DampingRatio - 0.2) / 0.1 \* (1.79 - 1.6) + 1.6;

B1 = (DampingRatio - 0.2) / 0.1 \* (1.63 - 1.5) + 1.5;

elseif 0.3 < DampingRatio && DampingRatio <= 0.4

BS = (DampingRatio - 0.3) / 0.1 \* (1.87 - 1.79) + 1.79;

B1 = (DampingRatio - 0.3) / 0.1 \* (1.7 - 1.63) + 1.63;

elseif 0.4 < DampingRatio && DampingRatio <= 0.5

BS = (DampingRatio - 0.4) / 0.1 \* (1.93 - 1.87) + 1.87;

B1 = (DampingRatio - 0.4) / 0.1 \* (1.75 - 1.7) + 1.7;

elseif 0.5 < DampingRatio

BS = 1.93;

B1 = 1.75;

end

B = B1;

DD = g / (4 \* pi ^ 2) \* SaD \* TeD ^ 2 / B;

*% syms Qd; % be careful performance issue.*

*% Kd = KeD - Qd / DD;*

*% Ku = Kd / postYieldingRatio;*

*% Dy = Qd / (Ku - Kd);*

*% ATD = 4 \* Qd \* (DD - Dy);*

*% solQd = double(solve(DampingRatio == ATD / (2 \* pi \* KeD \* DD ^ 2), Qd))*

Qd = dleta;

nextDampingRatio = 0;

while abs(nextDampingRatio - DampingRatio) > dleta

Kd = KeD - Qd / DD;

Ku = Kd / postYieldingRatio;

Dy = Qd / (Ku - Kd);

ATD = 4 \* Qd \* (DD - Dy);

nextDampingRatio = ATD / (2 \* pi \* KeD \* DD ^ 2);

Qd = Qd + dleta;

end

B

DD

KeD

*% Fy1 = Ku \* Dy / LRBs*

singleKu = Ku / LRBs

singleKd = Kd / LRBs

singleQd = Qd / LRBs

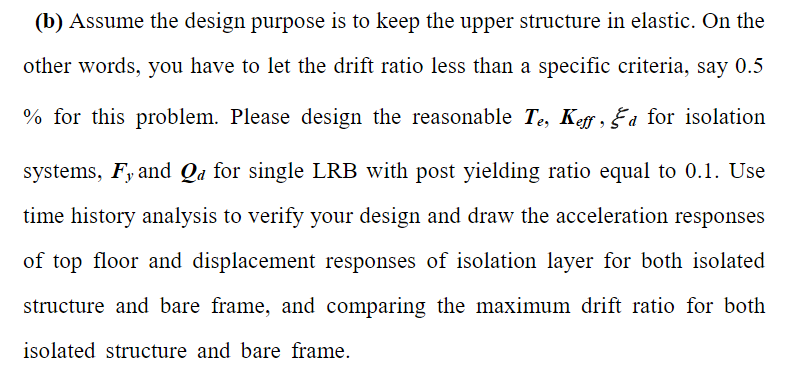
singleKeD = KeD / LRBs

DampingRatio

singleKu = Ku / LRBs

Fy = (Qd + Kd \* Dy) / LRBs

postYieldingRatio



假設初始設計目標：

* TeD = 3sec
* Damping Ratio = 20%
* Post Yielding Ratio = 0.1



* TeD => SaD = 0.195
* W = 35.53421364 (from ETABS)

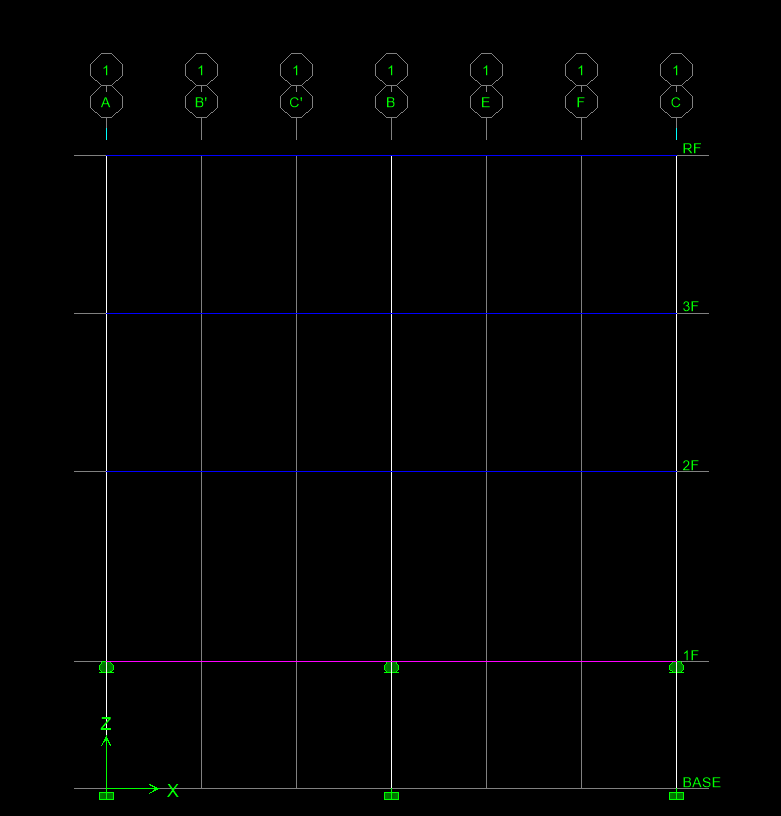
由以上資訊經迭代可得：

* Keff for isolation systems = 15.8889 tf/m
* Fy for single LRB = 0.1897 tf
* Qd for single LRB = 0.1707 tf

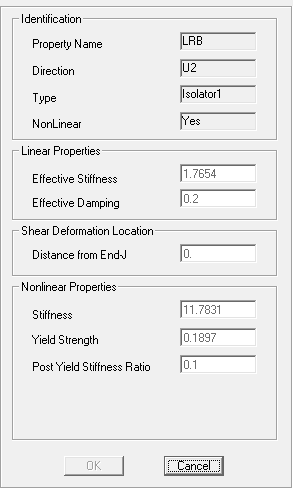
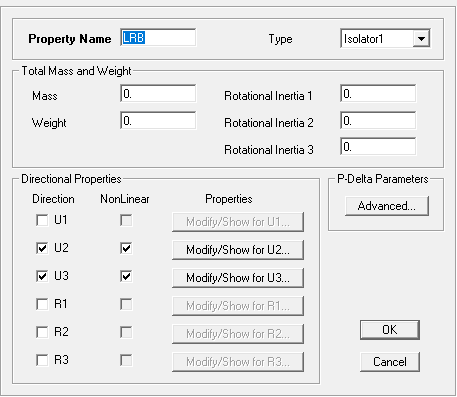
程式碼邏輯同第二題

接著進入模型，模擬 LRB 性質

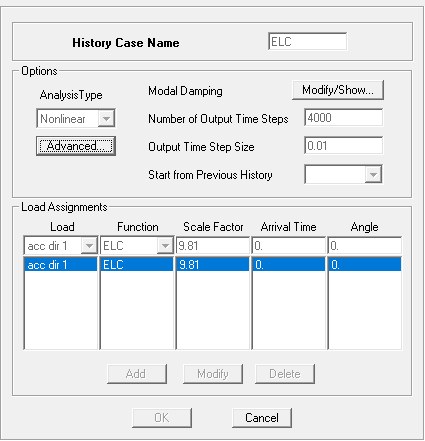
* 首先，底下加一層隔震墊。



* 設定 LRB 性質
* 由於 ETABS 9 U1 方向不能 fix 所以是在模型頂端設定 roller。



* 輸入正規化後的地震歷時



* 畫出頂樓加速度歷時



* 隔震墊位移歷時



* Maximum Drift Ratio



* 遲滯迴圈



clc; clear; close all;

periods = filename\_to\_array('DA\_BARE\_v1', 3, 1, 10);

acce\_bare = filename\_to\_array('DA\_BARE\_v1', 3, 3, 10);

disp\_iso = filename\_to\_array('DA\_ISO\_v3', 3, 2, 10);

acce\_iso = filename\_to\_array('DA\_ISO\_v3', 3, 3, 10);

force\_hysteresis = filename\_to\_array('hysteresis\_v2', 3, 2, 10);

disp\_hysteresis = filename\_to\_array('hysteresis\_v2', 3, 3, 10);

figure;

plot(periods, acce\_bare, periods, acce\_iso);

title('RF acceleration responses');

xlabel('T(sec)');

ylabel('acceleration(m/s^2)');

legend('bare frame', 'isolated structure', 'Location', 'northeast');

figure;

plot(periods, disp\_iso);

title('displacement responses of isolation layer');

xlabel('T(sec)');

ylabel('displacement(m)');

figure;

plot(disp\_hysteresis, force\_hysteresis);

title('hysteresis loop');

xlabel('displacement(m)');

ylabel('force(tf)');

axis([-0.15 0.35 -0.4 0.6]);

附錄：

function *data* = filename\_to\_array(*filename*, *total\_col*, *array\_col*, *headlines*)

*%*

*% 適用於被動控制給的 file\_input.*

*%*

*% @since 3.1.0*

*% @param {string} [filename] 檔案名稱.*

*% @param {number} [total\_col] 總共有幾欄.*

*% @param {number} [array\_col] 要回傳第幾欄.*

*% @param {number} [headlines] 忽略標題.*

*% @return {array} [data] 加速度歷時資料.*

*% @see ignore\_headlines*

*%*

if nargin == 3

headlines = 0;

end

fileID = fopen((filename + ".txt"), 'r');

ignore\_headlines(fileID, headlines);

repeat\_f = repmat('%f', 1, total\_col);

A = fscanf(fileID, repeat\_f, [total\_col Inf]).';

fclose(fileID);

data = A(:, array\_col);

end

function [] = ignore\_headlines(*fileID*, *headlines*)

*%*

*% 忽略標題.*

*%*

*% @since 1.0.0*

*% @param {number} [fileID] fileID.*

*% @param {number} [headlines] 要忽略的行數.*

*%*

for index = 1 : headlines

fgetl(fileID);

end

end

function [*u*, *v*, *a*] = newmark\_beta\_calculation(*m*, *c*, *k*, *up*, *vp*, *ap*, *dp*, *dt*, *gamma\_*, *beta\_*)

*%*

*% newmark beta calculation*

*%*

*% @since 1.0.0*

*% @param {number} [m] 質量.*

*% @param {number} [c] 阻尼.*

*% @param {number} [k] 勁度.*

*% @param {number} [up] 前一個點的位移.*

*% @param {number} [vp] 前一個點的速度.*

*% @param {number} [ap] 前一個點的加速度.*

*% @param {number} [dp] 位移.*

*% @param {number} [dt] 時間間隔.*

*% @param {number} [gamma\_] 常數.*

*% @param {number} [beta\_] 常數.*

*% @return {number} [u] 位移.*

*% @return {number} [v] 速度.*

*% @return {number} [a] 加速度.*

*%*

*%------------------*

*% Calculate effective stiffness*

*%------------------*

k\_s = k + gamma\_ / (beta\_ \* dt) \* c + 1 / (beta\_ \* (dt ^ 2)) \* m;

*%------------------*

*% Compute coefficients a and b*

*%------------------*

a = 1 / (beta\_ \* dt) \* m + gamma\_ / beta\_ \* c;

b = 1 / (2 \* beta\_) \* m + dt \* (gamma\_ / (2 \* beta\_) - 1) \* c;

*%------------------*

*% Calculate effective excitation and all responses*

*%------------------*

dp\_s = dp + a \* vp + b \* ap;

*%~~*

du = dp\_s / k\_s;

*%==*

dv = gamma\_ / (beta\_ \* dt) \* du - gamma\_ / beta\_ \* vp - dt \* (1 - gamma\_ / (2 \* beta\_)) \* ap;

da = 1 / (beta\_ \* (dt ^ 2)) \* du - 1 / (beta\_ \* dt) \* vp - 1 / (2 \* beta\_) \* ap;

*%~~*

u = up + du;

v = vp + dv;

a = ap + da;

end

function [*u*, *v*, *a*] = newmark\_beta(*ag*, *time\_interval*, *damping\_ratio*, *tn*, *method*)

*%*

*% newmark beta.*

*%*

*% @since 1.0.0*

*% @param {array} [ag] input 的地震力歷時資料.*

*% @param {number} [time\_interval] 時間間隔.*

*% @param {number} [damping\_ratio] 阻尼比.*

*% @param {number} [tn] 自然週期.*

*% @param {string} [method] 'average' or 'linear'.*

*% @return {array} [u] 位移.*

*% @return {array} [v] 速度.*

*% @return {array} [a] 加速度.*

*% @see newmark\_beta\_calculation*

*%*

*% average Acceleration Method*

gamma\_ = 1 / 2;

switch method

*% average Acceleration Method*

case 'average'

beta\_ = 1 / 4;

*% linear Acceleration Method*

case 'linear'

beta\_ = 1 / 6;

end

wn = (2 \* pi) / tn;

m = 1;

c = 2 \* damping\_ratio \* wn \* m;

k = (wn ^ 2) \* m;

p\_t = - m \* ag;

u = zeros(size(p\_t));

v = zeros(size(p\_t));

a = zeros(size(p\_t));

a(1) = 1 / m \* (p\_t(1) - c \* v(1) - k \* u(1));

ag\_length = length(ag);

for ag\_index = 2 : ag\_length

dp = p\_t(ag\_index) - p\_t(ag\_index - 1);

[u(ag\_index), v(ag\_index), a(ag\_index)] = newmark\_beta\_calculation(m, c, k, u(ag\_index - 1), v(ag\_index - 1), a(ag\_index -1), dp, time\_interval, gamma\_, beta\_);

end

a = a + ag;

end